



Universidad
Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Ingeniería Industrial

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

**DESARROLLO DE UNA APLICACIÓN
INFORMÁTICA SOBRE ANDROID DEL
DISEÑO DE UN DISPOSITIVO MECÁNICO**

Autor: Miguel Esteban Lineros

Tutor: Higinio Rubio Alonso

Leganés, 2016

Título: **Desarrollo de una aplicación informática sobre Android del diseño de un dispositivo mecánico.**

Autor: **Miguel Esteban Lineros**

Director: **Higinio Rubio Alonso**

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día..... de.....
de..... en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid,
acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



AGRADECIMIENTOS

Este trabajo fin de grado supone el cierre de una etapa maravillosa.

Me gustaría agradecer el apoyo que me ha aportado mi familia desde el primer momento, que sin ellos no habría llegado hoy a donde estoy.

Agradecer en especial a Higinio, por su confianza puesta en mí y su dedicación.





RESUMEN

El presente proyecto consiste en el diseño e implementación de una aplicación para dispositivos multiplataforma con sistema operativo Android capaz de calcular y representar mecanismos de leva plana de rotación con seguidor traslacional.

Para la consecución del objetivo principal se ha creado un modelo capaz de interpretar y calcular ciertos parámetros del diseño de levas, analizándolos para determinar la factibilidad del problema. Para ello, se hará uso del entorno de programación para Android, “AppInventor”, que permitirá crear una aplicación multiplataforma sencilla en su uso, interactiva y precisa técnicamente.

Para comprobar la corrección de los cálculos realizados por la aplicación se ha realizado un conjunto de pruebas de diferentes situaciones en el diseño de estos mecanismos y se han verificado con otros programas informáticos.

Se ha comprobado su correcto funcionamiento y operatividad en varias plataformas, tanto smartphones como tabletas.





ABSTRACT

This project involves the design and implementation of an informatic application for multiplatform dispositives with Android operative systems, capable to calculate and represent rotating planar cams with linear follower mechanism.

A model has been developed to achieve the main goal, competent to understand and calculate singular parameters of the cam design, analyzing them in order to verify the practicability of the problem. This will be achieved using the Android programming environment, “ApplInventor”, which enables completion of an interactive, homely and sightly multiplatform app.

In order to verify the workings and operations of the application, a set of tests in various situations have been carried out in the design of this mechanism. In addition, the results of our app have been compared with other external software.

The app has been tested on a variety of platforms both on smartphones and tablet.





Índice

1.	MOTIVACIÓN Y OBJETIVOS DEL PROYECTO	1
1.1	Ámbito y motivación	3
1.2	Objetivos	7
1.3	Fases del proyecto	8
1.4	Estructura de la memoria	9
2.	TEORÍA DE LEVAS	11
2.1	Introducción	13
2.2	Terminología en los mecanismos leva-palpador	14
2.3	Clasificación de mecanismos leva-palpador	16
	Tipo de movimiento del seguidor	16
	Tipo de leva	17
	Tipo de cierre o junta	18
	Tipo de seguidor	18
	Tipo de restricción de movimiento	19
	Tipo de programa de movimiento	20
2.4	Diagramas SVAJ	20
2.5	Ley fundamental de diseño de levas	22
2.6	Funciones más comunes	23
	Movimiento Armónico Simple	24
	Movimiento Cicloidal	26
	Polinómico 3-4-5	27
	Polinómico 4-5-6-7	28
	Velocidad constante	28
	Movimientos de medio periodo	29
	Movimiento Armónico Doble	32
	Movimientos compuestos	33
2.7	Dimensionamiento de la leva	37
	Ángulo de presión	37
	Radio de curvatura	40
2.8	Consideraciones prácticas de diseño	43
	¿Seguidor traslacional o rotacional?	43
	¿Cierre de forma o de fuerza?	43
	¿Seguidor de rodillo o de cara plana?	44
	¿Leva radial o axial?	46
	¿Lubricar o no lubricar?	46



3.	HERRAMIENTAS DE SOFTWARE PARA LA PROGRAMACIÓN	49
3.1	Introducción	51
3.2	Android	51
3.3	Eclipse y otros entornos de programación	54
3.4	ApplInventor	55
4.	METODOLOGÍA	59
4.1	Etapas de diseño	61
4.2	Elección de la herramienta para el desarrollo	62
	Introducción a ApplInventor	62
	Ventajas e inconvenientes de ApplInventor	66
4.3	Desarrollo del proyecto	67
	Referencias	68
	Diseño de la aplicación	69
	Requisitos estéticos	86
5.	ANÁLISIS DE RESULTADOS. PRUEBA DE FUNCIONAMIENTO DE LA APLICACIÓN	91
5.1	Prueba de funcionamiento	93
5.2	Verificación en SC Levas	114
6.	CONCLUSIONES Y TRABAJOS FUTUROS	117
6.1	Conclusiones	119
6.2	Trabajos futuros	120
7.	PRESUPUESTO	123
8.	BIBLIOGRAFÍA	127
8.1	Referencias bibliográficas consultadas	129
8.2	Otros recursos utilizados	130
8.3	Principales referencias a la web	130
	ANEXO I. Manual de la app “Cálculo de levass”	131
	Dos tramos, seguidor de cara plana	137
	Tres tramos, seguidor puntual	142
	ANEXO II. Aplicación de prueba: Calculadora	147



Índice de figuras

Figura 1. Martillo de forja movido por una leva	3
Figura 2. Smartphones y tablets	5
Figura 3. Móvil con apps	6
Figura 4. Fases del proyecto	8
Figura 5. Mecanismo leva-seguidor	13
Figura 6. Árbol de levas en motor de combustión interna	14
Figura 7. Terminología estándar de los mecanismos leva-seguidor	14
Figura 8. Mecanismo leva-seguidor rotacional	16
Figura 9. Mecanismo leva-seguidor traslacional	16
Figura 10. Mecanismo leva radial-seguidor	17
Figura 11. Mecanismo leva axial-seguidor	17
Figura 12. Mecanismo de leva globoidal	17
Figura 13. Mecanismo de leva con cierre por fuerza	18
Figura 14. Mecanismo de leva con cierre de forma	18
Figura 15. Sistema de levas conjugadas	18
Figura 16. Seguidor puntual	19
Figura 17. Seguidor de rodillo	19
Figura 18. Seguidor plano	19
Figura 19. Seguidor de hongo	19
Figura 20. Diagramas $SVAJ$	21
Figura 21. Tramo parcial en diagrama de desplazamientos	22
Figura 22. Movimiento armónico simple de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	25
Figura 23. Movimiento cicloidal de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	26
Figura 24. Movimiento polinómico 3-4-5 de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	27
Figura 25. Movimiento polinómico 4-5-6-7 de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	28
Figura 26. Movimiento semiarmónico de ascenso de V_0 nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm ..	29
Figura 27. Movimiento semiarmónico de ascenso de V_f nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm ..	30
Figura 28. Movimiento semicicloidal de ascenso de V_0 nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm ...	30
Figura 29. Movimiento semicicloidal de ascenso de V_f nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm ...	31
Figura 30. Movimiento armónico doble de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	32
Figura 31. Movimiento de ascenso de aceleración de onda senoidal modificada. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	33
Figura 32. Construcción de la función de aceleración senoidal modificada [Norton, 2009]	34
Figura 33. Movimiento de ascenso de aceleración de onda trapezoidal modificada. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm	35
Figura 34. Construcción de la función de aceleración trapezoidal modificada [Norton, 2009]	36
Figura 35. Ángulo de presión en leva con palpador de rodillo	37
Figura 36. Leva con palpador plano de cara inclinada	38



Figura 37. Fenómeno de rebaje	41
Figura 38. Fenómeno de cúspide	41
Figura 39. Sistema Audi Valvelift con uso de seguidores de rodillo	45
Figura 40. Sistema Android implantado en ordenador de a bordo de un coche	51
Figura 41. Versión Android 6.0 Marshmallow	52
Figura 42. Ventas smartphones 2015	53
Figura 43. Software Eclipse	54
Figura 44. Software Android Studio	55
Figura 45. Software Basic4Android	55
Figura 46. Software AppInventor	56
Figura 47. Categoría de bloques en AppInventor	57
Figura 48. Requisitos de diseño de la aplicación	61
Figura 49. Conexión al teléfono vía WiFi	63
Figura 50. Uso de emulador en el ordenador	63
Figura 51. Conexión al teléfono vía USB	63
Figura 52. Pantalla "Designer"	64
Figura 53. Pantalla "Blocks"	65
Figura 54. Aviso superación 10 pantallas	67
Figura 55. Software SC Levas	68
Figura 56. Pantalla principal SC Levas	68
Figura 57. Pantalla visualización sistema leva-seguidor	69
Figura 58. Organigrama de la aplicación	70
Figura 59. Uso de componentes CheckBox	71
Figura 60. Selección de tramos y tipo de seguidor	71
Figura 61. Algoritmos para seleccionar únicamente un CheckBox	72
Figura 62. Error de compilación del proyecto	73
Figura 63. Ejemplo ventanas emergentes de ayuda	74
Figura 64. Funciones de curvas y derivadas usadas en la aplicación	74
Figura 65. Detalle función armónica simple de desplazamiento	75
Figura 66. Márgenes en las gráficas cinemáticas	77
Figura 67. Configuración escalado	78
Figura 68. Pasos implementación "Ángulo de presión"	79
Figura 69. Ventana emergente al pulsar sobre "Guardar"	79
Figura 70. Compartir documentos	80
Figura 71. Origen de coordenadas en Canvas	81
Figura 72. Bloque para situar función correctamente sobre el componente Canvas	81
Figura 73. Indicación de scroll para pantalla "Gráficas"	82
Figura 74. Representación gráfica de la implementación de las ecuaciones de seguidor de cara plana.....	83
Figura 75. Representación leva plana rotacional con seguidor de cara plana.....	84



Figura 76. Representación base de datos en AppInventor	85
Figura 77. Material Design – Icons	86
Figura 78. Ejemplo iconos Google Design	87
Figura 79. Pruebas previas icono	87
Figura 80. Icono final Material Design	87
Figura 81. System Icon – Teoría de levas	88
Figura 82. Diseño System Icon en Adobe Illustrator	88
Figura 83. Diseño animación de leva en Adobe Illustrator	89
Figura 84. Pantalla “Home”	94
Figura 85. Pantalla “Selección de leva”	95
Figura 86. Pantalla “Configuración de tramos”	96
Figura 87. Pantalla “Gráficas”. 1ª Parte	97
Figura 88. Pantalla “Gráficas”. 2ª Parte	98
Figura 89. Pantalla “Gráficas”. 3ª Parte	99
Figura 90. Reajuste magnitudes geométricas	100
Figura 91. Modificación ángulo de presión y radio de curvatura	101
Figura 92. Pantalla “Representación de leva”	102
Figura 93. Guardado de gráficas	103
Figura 94. Exportar “datos.csv”	104
Figura 95. Compartir “datos.csv” vía Gmail	104
Figura 96. Detalle fichero “datos” en Excel	105
Figura 97. Gráfica de desplazamiento en Excel	106
Figura 98. Gráfica de velocidad en Excel	106
Figura 99. Gráfica de aceleración en Excel	107
Figura 100. Gráfica de sobreaceleración en Excel	107
Figura 101. Exportar “puntosleva.csv”	108
Figura 102. Compartir “puntosleva.csv” vía Gmail	108
Figura 103. Detalle fichero “puntosleva” en Excel	109
Figura 104. Leva representada en Excel	109
Figura 105. Diseño de leva en Autodesk Inventor	110
Figura 106. Modelo CAD planta	111
Figura 107. Modelo CAD	111
Figura 108. Impresión 3D	112
Figura 109. Equipo para impresión 3D	112
Figura 110. Levas en 3D de diferentes tamaños	113
Figura 111. Configuración sistema leva-seguir en SC Levas	114
Figura 112. Diagramas cinemáticos en SC Levas	115
Figura 113. Perfil de la leva en SC Levas	116
Figura 114. Mecanismo leva-seguir en SC Levas	116
Figura 115. Pantalla inicial “Cálculo de levas”	131



Figura 116. Pantalla “Home”	131
Figura 117. Pantalla “Teoría de levas”. 1ª Parte	132
Figura 118. Pantalla “Teoría de levas”. 2ª Parte	133
Figura 119. Pantalla “Teoría de levas”. 3ª Parte	134
Figura 120. Pantalla “Sobre App”. 1ª Parte	134
Figura 121. Pantalla “Sobre App”. 2ª Parte	135
Figura 122. Pantalla “Sobre App”. 3ª Parte	136
Figura 123. Selección de leva-seguidor plano	137
Figura 124. Configuración de leva de 2 tramos	138
Figura 125. Gráficas seguidor de cara plana. 1ª Parte	139
Figura 126. Gráficas seguidor de cara plana. 2ª Parte	140
Figura 127. Representación mecanismo leva-seguidor de cara plana	141
Figura 128. Selección de leva-seguidor puntual	142
Figura 129. Configuración de leva de 3 tramos	143
Figura 130. Gráficas seguidor puntual. 1ª Parte	144
Figura 131. Gráficas seguidor puntual. 2ª Parte	145
Figura 132. Representación mecanismo leva-seguidor puntual	146
Figura 133. Visualización de mesa de trabajo en “Designer”	147
Figura 134. Detalle de propiedades de componente Button	148
Figura 135. App “Calculadora”	149



Índice de tablas

Tabla 1. Costes de formación	125
Tabla 2. Costes de desarrollo	125
Tabla 3. Costes de informática	126
Tabla 4. Otros costes	126
Tabla 5. Resumen de costes	126
Tabla 6. Presupuesto final	126





1. MOTIVACIÓN Y OBJETIVOS DEL PROYECTO





1.1. ÁMBITO Y MOTIVACIÓN

Identificamos la leva como el elemento que obliga el movimiento del seguidor mediante contacto directo. Dicho movimiento es el resultado de un programa. De esta manera, se puede pensar en el sistema leva-seguidor como un dispositivo de información mecánica, donde el seguidor traducirá en algún tipo de movimiento esta rotación de la leva como parámetro de entrada.

No se tiene un conocimiento seguro del surgimiento de estos mecanismos, pero existen vestigios de antiguos instrumentos musicales que se basaban en pequeñas estructuras dentadas a modo de cuñas que fueron encontrados en reliquias de la Edad Paleolítica hace alrededor de 10.000 años atrás. Al parecer, éstas se utilizaban a modo de levas planas lineales que producían sonido en cada golpe.

De cualquier modo, no fue hasta el siglo XV d.c. cuando Leonardo Da Vinci utilizó todo su ingenio para desarrollar diferentes mecanismos basados en la transmisión de movimientos con sistemas leva-seguidor. Uno de sus inventos más reconocidos que utiliza este tipo de dispositivos es un martillo golpeador accionado por una leva, el cual era de gran ayuda para labores arduas en el forjado de herramientas.



Figura 1. Martillo de forja movido por una leva.

Sin duda alguna, los mecanismos leva-seguidor constituyen de forma simple, versátil, compacta y económica una manera de transmitir el movimiento, que a diferencia de otros mecanismos como los articulados le convierten en un sistema generador de funciones flexible y útil, como nos remarca el prestigioso profesor Robert L. Norton [1].



Con frecuencia podemos encontrar en cualquier dispositivo mecánico sistemas leva-seguidor. Son utilizados en toda clase de máquinas, desde motores de combustión interna hasta muchas de las máquinas empleadas en la fabricación de bienes de consumo, pasando por máquinas de afeitar e incluso lavadoras. Desde luego los sistemas leva-seguidor son de los más antiguos y a su vez de los más recurridos.

Cabe destacar una de las grandes ventajas que proporcionan estos mecanismos a los ingenieros, y es que son dispositivos verdaderamente sencillos de diseñar para producir una función de salida específica. A día de hoy, existen diferentes programas computacionales tipo CAD/CAM para levas, que ofrecen grandes simplificaciones en el diseño y fabricación de conjuntos leva-seguidor y que trataremos más adelante.

De esta manera, y gracias al avance de las tecnologías, tenemos cada vez más fácil el acceso al diseño de elementos mecánicos. Pero la tecnología se desarrolla, tanto para bien como para mal, a una velocidad vertiginosa.

Un ejemplo claro del avance brutal de la tecnología lo tenemos en el sector de las telecomunicaciones y el mundo de la telefonía móvil. Resulta espectacular pensar que bien entrados en la segunda mitad del siglo pasado, Martin Cooper, directivo de Motorola realizó la primera llamada desde un teléfono móvil, y ahora este pequeño aparato se ha convertido en un inseparable compañero.

El desarrollo de estos terminales a través de diferentes generaciones ha traído consigo grandes avances en la tecnología, pues la adaptación a estos cambios es inevitable. Lo que conocemos hoy como teléfonos inteligentes o smartphones era, hace unos años, impensable.

Táctiles, multitarea, con acceso a internet vía WiFi o red 4G, GPS, sincronización con otros dispositivos, con grandes funciones de multimedia... los Smartphone son a día de hoy minicomputadoras con la apariencia y las funciones de un teléfono móvil. Ya no solo el crecimiento de estos aparatos es inmenso, sino que todo lo que le rodea y gobierna debe crecer a su mismo ritmo. Hablo de los sistemas operativos, que según datos del tercer trimestre de 2013 en cuanto a su uso en teléfonos inteligentes en todo el mundo [18], estos fueron los resultados:

- Android (de Google): 81.9%
- iOS (de Apple): 12.1 %
- Windows Phone (de Microsoft): 3.6%
- BlackBerry OS (de BlackBerry): 1.8%
- Bada (de Samsung): 0.3%
- Symbian OS (de Nokia): 0.2%
- Firefox OS (de Mozilla): 0.2%
- Otros: 0.2%



Figura 2. Smartphones y tablets.

Cabe destacar la gran diferencia entre Android e iOS, que se hace aún mayor cuando los estudios de un año anterior, en 2012, posicionaban a Android con un 72.4% y a iOS en un 13.9% en su uso a nivel mundial. Esto se puede deber a una gran lista de ventajas que presenta la primera plataforma respecto a la segunda y que trataremos más adelante.

Por otro lado, junto con el desarrollo de los smartphones, va creciendo en paralelo el uso y lanzamiento de las tablets, dispositivos de mayor tamaño con un rendimiento superior a los teléfonos inteligentes. Cada vez más, se están convirtiendo en herramientas de uso cotidiano que podemos encontrar en casa o en el trabajo, debido a su mayor facilidad de uso con respecto a un ordenador de sobremesa o portátil. Por esto no sorprende ver estudios publicados en páginas tecnológicas de prestigio como Xataka.com donde indiquen que las ventas de tablets estimadas para 2015 fuesen de 321 millones de unidades a nivel mundial, predominando también Android como la plataforma más solicitada por los usuarios [17].

Pero este no es sólo un mercado inmenso para usuarios y fanáticos de la tecnología, sino también para empresas desarrolladoras o autónomos que buscan explotar al máximo los recursos que ofrecen estos dispositivos mediante aplicaciones, o comúnmente conocidas como apps.



Figura 3. Móvil con apps.

Una app es un pequeño programa que se instala en nuestros smartphones o tablets con la finalidad de resolver una tarea o entretener al usuario. Podemos encontrar desde editores de texto, hojas de cálculo, libros, juegos... Pero lo que sin duda podemos apreciar de esta importante nueva forma de comunicación es la versatilidad y la globalidad que ha alcanzado en tiempo récord.

Así pues, la gran motivación de este proyecto es realizar un trabajo real dentro del campo de la ingeniería mecánica e informática que se perfila de gran importancia en un futuro inminente, demostrando que esta área de conocimiento puede tener alcance mundial adaptado a los nuevos tiempos. Este proyecto se fundamenta en el diseño e implementación de una aplicación multiplataforma sobre Android, fácil y accesible que cubra una aproximación al estudio y diseño de mecanismos de levas rotatorias planas con seguidor traslacional.



1.2. OBJETIVOS

El objetivo principal de este Trabajo Fin de Grado queda planteado como el *“diseño e implementación de una aplicación informática multiplataforma, soportada por smartphones y tablets, sobre el sistema operativo Android, para el estudio y diseño de levas rotatorias planas con seguidor traslacional”*.

La consecución de este proyecto lleva asociados unos objetivos o hitos intermedios necesarios para llevar a buen término el objetivo principal, los cuales se detallan a continuación:

- Recopilación de información de cómo desarrollar y crear aplicaciones Android.
- Búsqueda de un software genérico en el que implementar la aplicación.
- Diseñar una aplicación de prueba (con el software AppInventor).
- Creación de un modelo capaz de interpretar y calcular sistemas leva-seguidor, a partir de entrada de datos.
- Diseño de una aplicación de elevado nivel técnico pero a su vez de manejo sencillo y amigable.
- Cuidado de la estética en el diseño de la aplicación, con la adaptación a las pautas proporcionadas por Google.
- Implementación del modelo en AppInventor.
- Realizar un manual de uso de la aplicación “Cálculo de levas”.
- Diseño de juego de pruebas (para varios casos posibles) que verifiquen la correcta operatividad de la aplicación creada.

En cuanto a los medios empleados para la realización de este Trabajo Fin de Grado se emplearán una serie de programas informáticos:

- El programa AppInventor se usará como plataforma de desarrollo de programación de nuestra aplicación Android. AppInventor es un entorno de desarrollo abierto, integrado vía web multiplataforma que permite su uso sobre soporte Java, de amplia utilización para desarrollar apps. Realmente, la selección del programa de desarrollo, entre los diferentes del mercado, ya sería parte del proyecto.
- El software Adobe Illustrator se empleará para diseñar el logo, las animaciones y los botones de acción más importantes que contenga nuestra app.
- Además, se empleará el software Ms-Office: la hoja de cálculo Excel para realizar los cálculos de geometría de las levas y compararlos con los resultados de la aplicación y el Word para la redacción del informe.



1.3. FASES DEL PROYECTO

En un proyecto de ingeniería genérico hay que definir unas fases de desarrollo para la consecución del objetivo perseguido. Los proyectos de innovación permiten una mayor libertad a la hora de llevarlos a cabo, ya que no establecen ninguna restricción previa para cumplir los objetivos del proyecto. La primera parte del mismo es de diseño y por tanto libre imaginación y originalidad de quien lo realiza, mientras que la segunda parte y principal se basa en datos e interpretaciones donde habrá que realizar comparaciones y sacar conclusiones.

Para la consecución del objetivo se abordaron las siguientes fases:

- Fase I: Estudios previos.

En esta primera fase plantearemos los objetivos, motivaciones y necesidades que la aplicación debe cumplir. Se realiza un diseño conceptual y sobre él se hacen los estudios de viabilidad funcional.

- Fase II: Diseño y análisis.

En la segunda fase del proyecto se modela el diseño de la aplicación para la simulación y obtención de datos. Después, los datos serán analizados y se valorará si hay que realizar modificaciones sobre el modelo preliminar de la aplicación.

- Fase III: Implementación final.

Por último, si los resultados del modelo son satisfactorios y cumplen con los objetivos establecidos, se procederá a implementar el diseño final, el cuál será sometido a pruebas oportunas que determinen su validez y correcto funcionamiento.

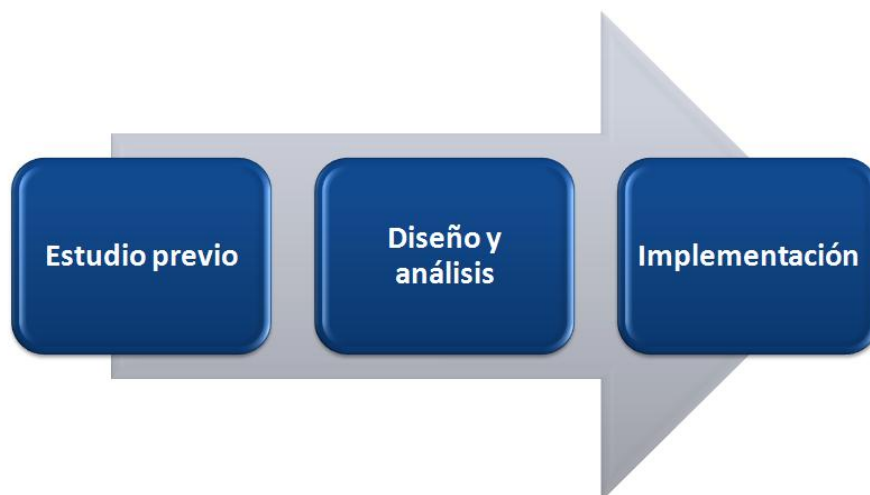


Figura 4. Fases del proyecto.



1.4. ESTRUCTURA DE LA MEMORIA

La estructura de este Trabajo Fin de Grado queda finalmente dividida en 8 capítulos, incluyendo bibliografía, que se describen a continuación:

- **Capítulo I** - Motivación y objetivos del proyecto.

En este primer capítulo se expondrán las motivaciones y objetivos del proyecto. Se explicarán las distintas fases que se han llevado a cabo para la consecución del mismo y por último, se introducirán los principales temas de los que consta la estructura de la memoria.

- **Capítulo II** - Teoría de levas.

Este capítulo tratará de profundizar en aspectos teóricos importantes para el cálculo y diseño de levas. Se expondrá desde una detallada clasificación hasta los parámetros más importantes a tener en cuenta en el estudio de estos mecanismos.

- **Capítulo III** - Herramienta de software para la programación.

El tercer capítulo de la memoria versará sobre el sistema operativo seleccionado, Android, y algunas de las herramientas existentes en el mercado para el desarrollo de aplicaciones en él. Ahondaremos en algunas de las características de la herramienta seleccionada: AppInventor.

- **Capítulo IV** - Metodología.

El capítulo de la metodología explicará fundamentalmente las pautas que se deben seguir para la creación de nuestra aplicación. También se incluirán en él algunas de las grandes ventajas que presenta nuestra plataforma de desarrollo frente a otras, y sus limitaciones.

- **Capítulo V** - Análisis de resultados. Prueba de funcionamiento de la aplicación.

En este capítulo se propondrá un ejercicio resuelto que demostrará el correcto funcionamiento de la aplicación. Se buscará resaltar la funcionalidad de la aplicación y el potencial que se ha alcanzado con el procesamiento de datos.

- **Capítulo VI** - Conclusiones y trabajos futuros.

El capítulo seis mostrará las conclusiones que se han alcanzado con el Trabajo Fin de Grado. Se propondrán trabajos futuros de mejora y evolución de la aplicación resaltando el gran potencial que presenta esta área del conocimiento a día de hoy y de mañana.



- **Capítulo VII - Presupuesto.**

Se ofrece una estimación de los costes de realización de la aplicación.

- **Capítulo VIII – Bibliografía.**

En el capítulo de bibliografía se exponen y enumeran los principales documentos bibliográficos y electrónicos consultados durante la realización del presente proyecto.

- **Anexo I - Manual de “Cálculo de levass”.**

Este capítulo adicional servirá al lector un manual de usuario con todas las pantallas disponibles de la aplicación. Resultará un capítulo útil para el correcto entendimiento y uso de la aplicación, así como descubrir el alcance de la misma.

- **Anexo II - Aplicación de prueba: Calculadora.**

En este adjunto a la memoria explicaremos una aplicación de prueba realizada para estudiar el poder computacional de cálculo de nuestro entorno de programación. También servirá como primera toma de contacto para observar la magnitud operativa de ApplInventor y ver qué capacidad tiene para aplicaciones de cálculo importantes.



2. TEORÍA DE LEVAS





2.1. INTRODUCCIÓN

Una leva es un elemento mecánico que sirve para impulsar a otro, por contacto puntual o lineal, llamado palpador o seguidor, para que este desarrolle una ley de movimiento especificada.

Un mecanismo leva-seguidor consiste en dos miembros móviles, cada uno de ellos con un grado de libertad sin coaccionar, en contacto mediante un par superior y ambos unidos a un miembro fijo o bastidor a través de un par de orden inferior.

De esta manera se obtiene un mecanismo de un sólo grado de libertad donde se establece una ley de dependencia entre los movimientos descritos por leva y palpador. A esta ley se le denomina usualmente ley de desplazamiento del palpador o ley fundamental del diseño de levas.

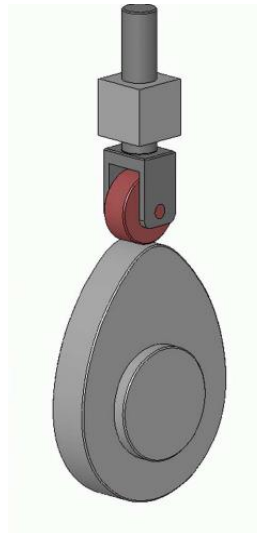


Figura 5. Mecanismo leva-seguidor.

Los sistemas leva-seguidor están presentes en cualquier lugar. Son utilizados en una amplia gama de dispositivos y mecanismos, lo que hace impensable imaginar un mundo industrializado a día de hoy sin estos sistemas. El movimiento de la leva se transforma en un movimiento de rotación o traslación, el cual se aprovecha para interesantes aplicaciones.

Dichos mecanismos son comúnmente utilizados para la transmisión del movimiento para las aperturas de las válvulas en motores de combustión interna. Están presentes en máquinas del sector textil y fabricación de cualquier bien, desde lavadoras a máquinas de afeitar.

Estos sistemas generalmente son utilizados como generadores de función. Son sencillos y poco costosos, tienen pocas piezas móviles y ocupan espacios muy reducidos. No son difíciles de diseñar, mostrando una gran flexibilidad y variedad en su conformado. Además, presentan la ventaja de un mantenimiento fácil y una fabricación en grandes volúmenes si fuese necesario.

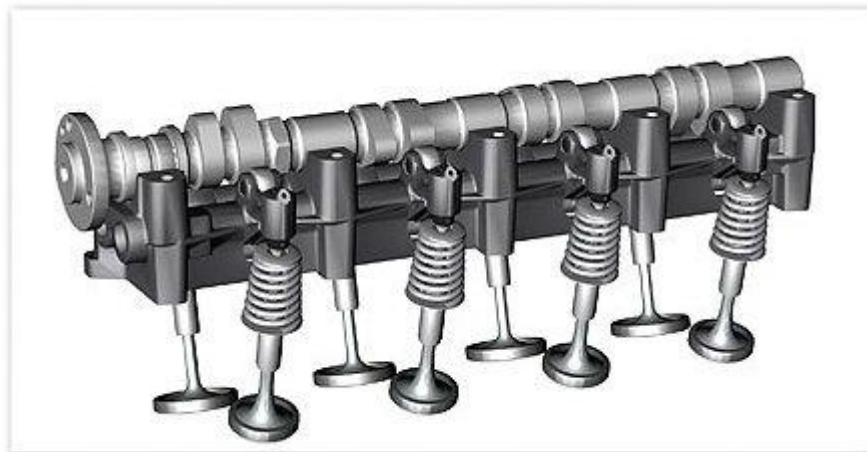


Figura 6. Árbol de levas en motor de combustión interna.

2.2. TERMINOLOGÍA EN LOS MECANISMOS LEVA-PALPADOR

En la siguiente imagen se muestra un mecanismo de leva plana con palpador de rodillo que servirá para explicar cada uno de los parámetros más importantes que podemos encontrar en estos mecanismos y describir la terminología utilizada en este trabajo.

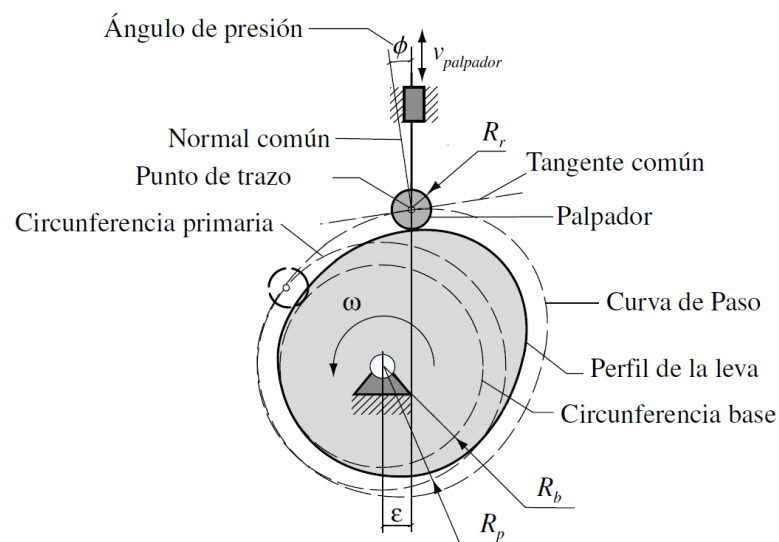


Figura 7. Terminología estándar de los mecanismos leva-seguidor.



La terminología utilizada en los mecanismos leva-palpador es la siguiente:

- Circunferencia base: Es la circunferencia más pequeña, de radio R_b , que puede trazarse con centro en el eje de rotación de la leva y tangente a la superficie física de ésta. En el caso de un palpador de rodillo es más pequeña que la circunferencia primaria, donde descontaremos el radio del rodillo R_r . Resulta un parámetro importante para el buen diseño de nuestra leva y su correcto funcionamiento.
- Circunferencia primaria: Es la circunferencia más pequeña, de radio R_p , que se puede trazar con centro en el eje de rotación de la leva y tangente a la curva de paso. Sólo se aplica en el caso de palpadores de rodillo.
- Curva de paso: Es la trayectoria que describe el centro del rodillo en la referencia solidaria a la leva, al completarse una vuelta de ésta. Corresponde a una curva offset del perfil de la leva (perfil teórico), separada una distancia igual al radio del rodillo R_r . En el caso de un palpador puntual ($R_r = 0$), esta curva coincide con el perfil de la leva.
- Punto de trazo: Es un punto del palpador que al realizar la inversión cinemática describe la trayectoria que constituye el perfil real de la leva (cuando el palpador es puntual), o el perfil teórico de la leva (cuando el palpador es circular).
- Ángulo de presión: Es el ángulo formado entre la normal común a los perfiles de la leva y del palpador en el punto geométrico de contacto y la dirección de la velocidad de dicho punto del palpador. Si el palpador es de rodillo debe considerarse la dirección de la velocidad de su centro. Este parámetro lo estudiaremos con detenimiento más adelante para el correcto dimensionamiento de la leva.
- Excentricidad: Es la distancia entre el eje a lo largo del cual se traslada el palpador y el centro de rotación de la leva. Su valor es nulo para palpadores axiales o alineados. Sólo está presente en palpadores con movimiento de traslación. La modificación de su valor puede corregir el efecto del ángulo de presión entre leva y palpador y conducir a un mecanismo adecuado.



2.3. CLASIFICACIÓN DE MECANISMOS LEVA-PALPADOR

Podemos encontrar distintas clasificaciones de los sistemas leva-seguidor: por el tipo de movimiento del seguidor, por el tipo de leva, por el tipo de cierre o junta, por el tipo de seguidor, por el tipo de restricciones de movimiento, por el tipo de programa de movimiento... A continuación analizamos cada uno de estas categorías con detenimiento.

Tipo de movimiento del seguidor

- Seguidor rotacional
- Seguidor trasladante o traslacional

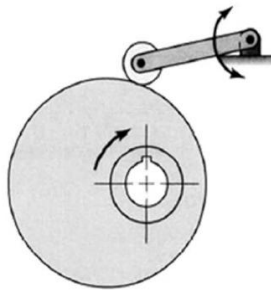


Figura 8. Mecanismo leva-seguidor rotacional.

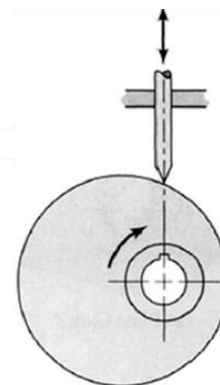


Figura 9. Mecanismo leva-seguidor traslacional.

La elección entre estos tipos de leva-seguidor es normalmente determinada por el tipo de movimiento de salida deseado. Si se requiere de una traslación rectilínea real, se elige el seguidor trasladante. Por otro lado, para un movimiento oscilatorio la elección es obvia.

Desde el libro de estudio “Diseño de maquinaria” de Robert L. Norton [1], el profesor hace diferentes comparaciones entre este tipo de mecanismos y sistemas de mecanismos de cuatro barras articuladas, asemejando mecanismos de manivela-balancín con el primer caso y de manivela-corredera con el segundo.

Un requisito fundamental que solemos encontrar en la industria es el tamaño del mecanismo, ya que en muchas ocasiones disponemos de espacios reducidos donde incorporar nuestro sistema. Más adelante veremos cuál es el más utilizado por cuestiones como esta y cómo se puede hacer un buen dimensionamiento que cumpla con requisitos de fluidez en el movimiento entre otros, distinguiendo entre estas dos configuraciones de mecanismos leva-seguidor.



Tipo de leva

La clasificación en este caso depende de la dirección del movimiento relativo del seguidor respecto al eje de rotación de la leva. Con normalidad distinguimos levas radiales o axiales.

- Leva radial
- Leva axial

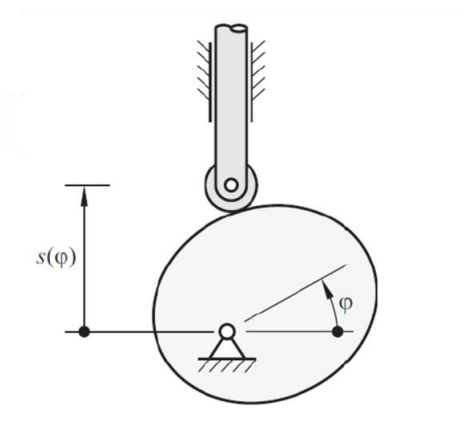


Figura 10. Mecanismo leva radial-seguidor.

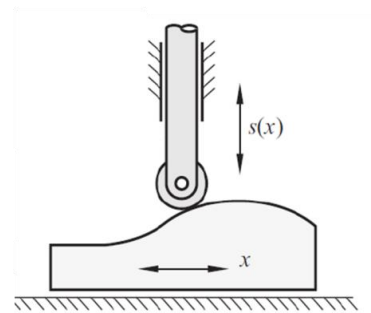


Figura 11. Mecanismo leva axial-seguidor.

Cabe destacar las levas tridimensionales o globoides, siendo una combinación de levas radial y axial. Forman un sistema de dos grados de libertad donde el movimiento del seguidor es una función de ambas entradas: la rotación de la leva con respecto a su eje y la traslación de la leva a lo largo de su eje. Empleadas cuando la masa del seguidor es muy grande.

Es fácil encontrar este tipo de mecanismo en indexadores mecánicos. Unidades mecánicas de ejes paralelos o cruzados que transforman la rotación uniforme del eje de entrada en rotación intermitente del eje de salida.



Figura 12. Mecanismo de leva globoidal.



Tipo de cierre o junta

- De fuerza: necesidad de elemento que ejerza fuerza externa a la junta para mantener a los dos eslabones en contacto en todo momento. Esta fuerza es proporcionada por un resorte.
- De forma: cierre de la junta por geometría. No se requiere ninguna fuerza externa.
- De leva conjugada: dos levas fijas sobre un eje común que son conjugados matemáticos entre sí. A su vez, dos seguidores conectados a un brazo común son empujados en direcciones opuestas por las levas conjugadas.

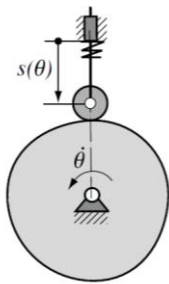


Figura 13. Mecanismo de leva con cierre por fuerza.

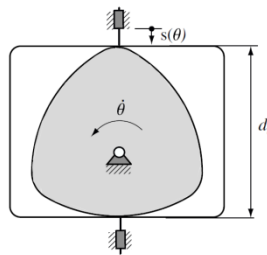


Figura 14. Mecanismo de leva con cierre de forma.

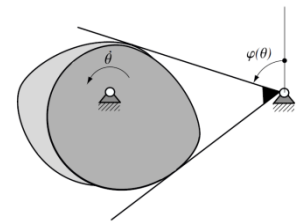


Figura 15. Sistema de levas conjugadas.

Cabe mencionar que cuando se utilizan levas con cierre de forma en trenes de válvulas de un motor de motocicleta o de automóvil se denominan levas desmodrómicas.

Tipo de seguidor

En este caso nos referimos a la parte del seguidor que está en contacto con la leva. Los más comunes son:

- De cuña o puntual
- De rodillo
- De cara plana
- De hongo

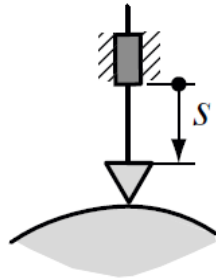


Figura 16. Seguidor puntual.

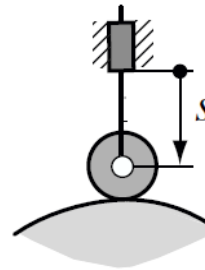


Figura 17. Seguidor de rodillo.

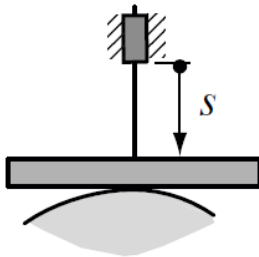


Figura 18. Seguidor plano.

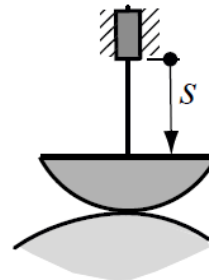


Figura 19. Seguidor de hongo.

En el primer caso, como su nombre indica, consiste en un borde afilado en contacto con la leva. Aunque su construcción resulta verdaderamente sencilla, este tipo de seguidores no son del todo útiles en la práctica por el excesivo desgaste del punto de contacto.

El seguidor de rodillo tiene la ventaja de poseer menor fricción, a diferencia del contacto deslizante de los otros, pero es más costoso. Se utilizan con frecuencia en maquinaria de producción por su facilidad de reemplazo. En levas de ranura donde el tipo de cierre es por geometría se suelen utilizar estos tipos de seguidores.

Los seguidores de cara plana son por lo general más pequeños que los anteriores en algunos diseños de leva por lo que usualmente se prefieren, así como por su menor coste. Los seguidores de hongo y de cara plana se diseñan y fabrican sobre pedido para cada aplicación.

Tipo de restricciones de movimiento

- Posición crítica extrema (CEP)
- Movimiento de trayectoria crítica (CPM)

La posición crítica extrema se refiere al caso en que las especificaciones de diseño definen las posiciones inicial y final del seguidor (posiciones extremas), pero no especifican ninguna



restricción de movimiento entre las posiciones extremas. Es fácil de plantear ya que el diseñador tiene la libertad de elegir las funciones de la leva que controlan el movimiento entre los extremos.

Movimiento de trayectoria crítica es un problema más restringido ya que el movimiento y/o una o más de sus derivadas se definen en todas o en una parte del intervalo de movimiento. Es decir, se especifican las posiciones inicial y final además de la trayectoria entre ellas. La definición de la ley de movimiento la impone la tarea tecnológica a realizar.

Tipo de programa de movimiento

Durante el ciclo de movimiento de la leva, el palpador realiza una serie de eventos que clasificamos en alejamientos (S), acercamientos (B) y detenciones (D). El alejamiento S es el movimiento que realiza el palpador al alejarse del centro de rotación. Por consiguiente el acercamiento B es el movimiento del palpador al acercarse al centro de rotación, y el detenimiento D es el evento de movimiento durante el cual el palpador permanece a una misma distancia del centro de rotación de la leva mientras esta gira.

La secuencia de estos eventos durante una vuelta completa de la leva determina un programa de movimiento para el palpador. De acuerdo con lo planteado en la bibliografía consultada [1] existen tres principales programas de movimiento para el palpador:

- Alejamiento – Acercamiento (S-B): Este es el programa más simple de todos, donde el seguidor solo ascendería y descendería. Adecuado para levas con palpador excéntrico.
- Alejamiento – Acercamiento – Detenimiento (S-B-D): Programa con un único detenimiento tras los eventos de alejamiento y acercamiento.
- Alejamiento – Detenimiento – Acercamiento – Detenimiento (S-D-B-D): En este caso se introduce un detenimiento entre los eventos de alejamiento y acercamiento.

2.4. DIAGRAMAS SVAJ

La primera tarea con la que se debe enfrentar el diseñador de levas es la selección de funciones matemáticas que definirán el perfil de la leva y por consiguiente el movimiento del seguidor. La aproximación más fácil para esto es “linealizar” la leva, esto es, “desenrollarla” de su forma circular y considerarla como una función graficada en ejes cartesianos.

Esta función graficada será la función de desplazamiento S , su primera derivada la función de velocidad V , la tercera será la función de aceleración A , y la cuarta la función de golpeteo o sobreaceleración J . Todas en ejes alineados representando en el eje horizontal el movimiento angular de la leva.

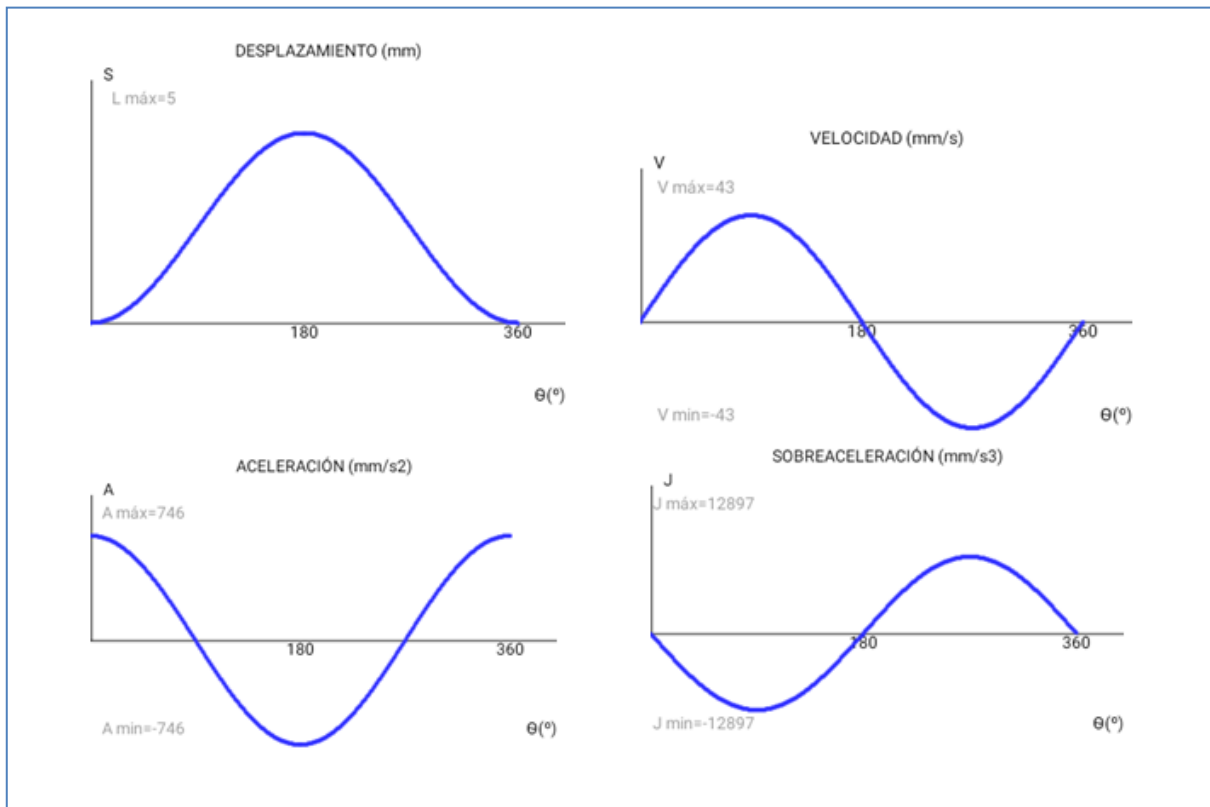


Figura 20. Diagramas S V A J.

En las figuras anteriores podemos ver un claro ejemplo de los cuatro diagramas SVAJ. En este caso, no existe discontinuidad en ninguno de ellos, eso tiene unas consecuencias positivas en el diseño de la leva, como veremos más adelante.

Parándonos a mirar el primero de los diagramas, el de desplazamiento, vemos que es la traducción de los recorridos realizados por el seguidor durante un ciclo (360°), es decir la función desplazamiento. Esta función se construye por la unión de diferentes tipos de tramos parciales.

En levas planas rotativas, cada intervalo se trata de manera independiente, definiendo una variable adimensional como cociente de dos ángulos: el numerador, θ , corresponde al ángulo parcial recorrido dentro del intervalo y el denominador, β , es el ángulo total de rotación del intervalo. Cuando definamos las funciones más comunes en el diseño de levas daremos cuenta de ello. Además del tipo de función a desarrollar y el ángulo parcial de la rotación se ha de indicar la elevación del seguidor, para definir completamente el movimiento, y la velocidad angular de la leva.

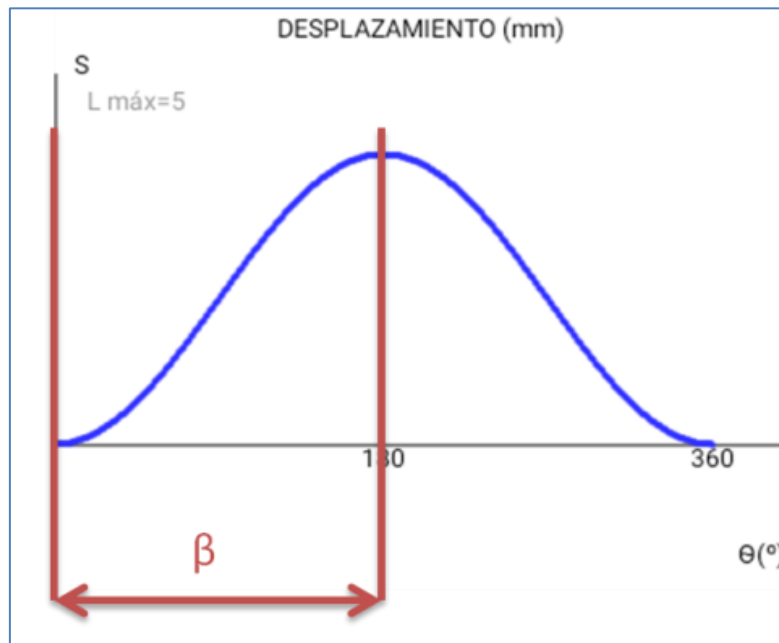


Figura 21. Tramo parcial en diagrama de desplazamientos.

2.5. LEY FUNDAMENTAL DEL DISEÑO DE LEVAS

Como se ha mencionado, la continuidad en los diagramas de desplazamiento y sus derivadas trae consigo aspectos positivos del resultado de diseño de una leva.

En el diseño de una ley de desplazamiento para el palpador existen especificaciones obligatorias y otras recomendables. La elección de las especificaciones no impuestas por la funcionalidad es una tarea importante para el diseñador, pues la ley de desplazamiento resultante determinará además del contorno de la leva, la cinemática del palpador y, por tanto, la dinámica del mecanismo.

Esto nos lleva a definir la ley fundamental de diseño de levas según R. L. Norton [1]:

La función de la leva debe ser continua en la primera y segunda derivadas de desplazamientos a través de todo el intervalo (360 grados). La función de rapidez de aceleración o sobreaceleración debe ser finita a través de todo el intervalo (360 grados).

Es imprescindible que la velocidad del palpador sea una función continua, de lo contrario se originarían aceleraciones teóricamente infinitas, y fuerzas elevadas que en máquinas de velocidad considerable podrían conducir a la destrucción del mecanismo o a la pérdida del contacto leva-palpador cuando el cierre del par es por fuerza.

De la misma manera las aceleraciones deben ser funciones de gráficas continuas para evitar variaciones importantes en las fuerzas que actúan en el mecanismo, a la vez que disuadir



ruidos y vibraciones que darían lugar a roturas, fatiga, etc., y a un desajuste en el movimiento real del palpador y el requerido.

Podemos aceptar discontinuidades en la tercera derivada (teniendo en cuenta que esto puede provocar vibraciones que se verán agravadas al aumentar la velocidad de giro de la leva por lo que, en el caso de velocidades angulares elevadas, no debe hacerse tal excepción).

No hay que dejar de pasar por alto que el movimiento de una leva es una situación cíclica, por lo que la continuidad debe estar asegurada también entre el punto inicial y su unión con el punto final (que son en sí el mismo punto).

2.6. FUNCIONES MÁS COMUNES

En este apartado describiremos las curvas más típicas utilizadas para el desarrollo del perfil de la leva. Dichas curvas se utilizan para la construcción de la ley de desplazamiento del seguidor, y junto a ella las leyes de velocidad, aceleración y sobreaceleración. Se busca presentar éstas al lector y mostrar algunas de sus características para la generación del perfil, con ejemplos extraídos del software de referencia, SC Levas. No se entrará en profundo detalle.

Antes de mostrar las curvas, cabe recordar la forma en la que se opera con cada función. Como hemos visto, cada ley de desplazamiento se completa por la unión de diferentes tramos que cumplan las condiciones de contorno requeridas. Se tratará cada intervalo de forma independiente dentro de sus límites de rotación. En apartados anteriores definimos Θ como el ángulo parcial recorrido dentro del intervalo y β como el ángulo total de rotación del intervalo. De esta forma, el cociente Θ/β varía de 0 a 1, siendo el 0 el punto inicial y 1 el punto final del tramo.

La curva de desplazamiento será muy parecida para casi todos los movimientos, sin embargo, encontraremos notables diferencias en sus derivadas. Las condiciones de contorno y los valores pico de las mismas no tendrán porqué ser iguales. Esta observación será el primer criterio que debe tener el diseñador para juzgar si su leva es aceptable o no. En primer lugar debe caer en la cuenta de la continuidad, como mínimo en velocidad y aceleración, y después buscar alternativas de movimientos con objeto de reducir los valores pico de estos diagramas.

Una vez visto esto, se han de asegurar otros parámetros y magnitudes que harán de nuestro perfil uno aceptable. En estos casos se tendrán en cuenta geometrías de leva y seguidor, así como los movimientos programados. En caso de obtener un perfil que no sea suficientemente bueno se tendrán que recalcular estos parámetros para alterar la solución anterior, como veremos más adelante.



Las curvas a analizar son las siguientes:

- Armónico Simple.
- Cicloidal.
- Polinómico 3-4-5 (caso particular de polinómico de 5º grado).
- Polinómico 4-5-6-7 (caso particular de polinómico de 7º grado).
- Velocidad constante.
- Movimientos de medio periodo (semiarmónicos y semicicloidales).
- Armónico Doble.
- Movimientos compuestos (Aceleración de Onda Senoidal Modificada y Aceleración de Onda Trapezoidal Modificada).
- Detenimiento.

Movimiento Armónico Simple

Las funciones armónicas tienen la propiedad de permanecer continuas a través de cualquier número de diferenciaciones. La constante transformación de senos y cosenos según avanzamos en sus derivadas hace de este tipo de movimiento muy recurrido para programas simples, de ascensos y descensos consecutivos, donde la ley de la continuidad se cumplirá siempre.

La principal peculiaridad de esta curva es que no se puede combinar con detenimientos, pues en éstos, al permanecer inmóvil el seguidor durante el giro de la leva, su velocidad, aceleración y sobreaceleración son cero. De esta manera aparecerían discontinuidades en aceleración, teniendo que rechazar el perfil de leva por incumplimiento de la ley de la continuidad.

Como aclaración, aunque se hable de ascenso o descenso, estas curvas son tan aplicables a seguidores traslacionales como oscilantes. En este último caso, la subida se tomará como un giro en el sentido de las agujas del reloj.



Sus ecuaciones para un movimiento de subida son las siguientes:

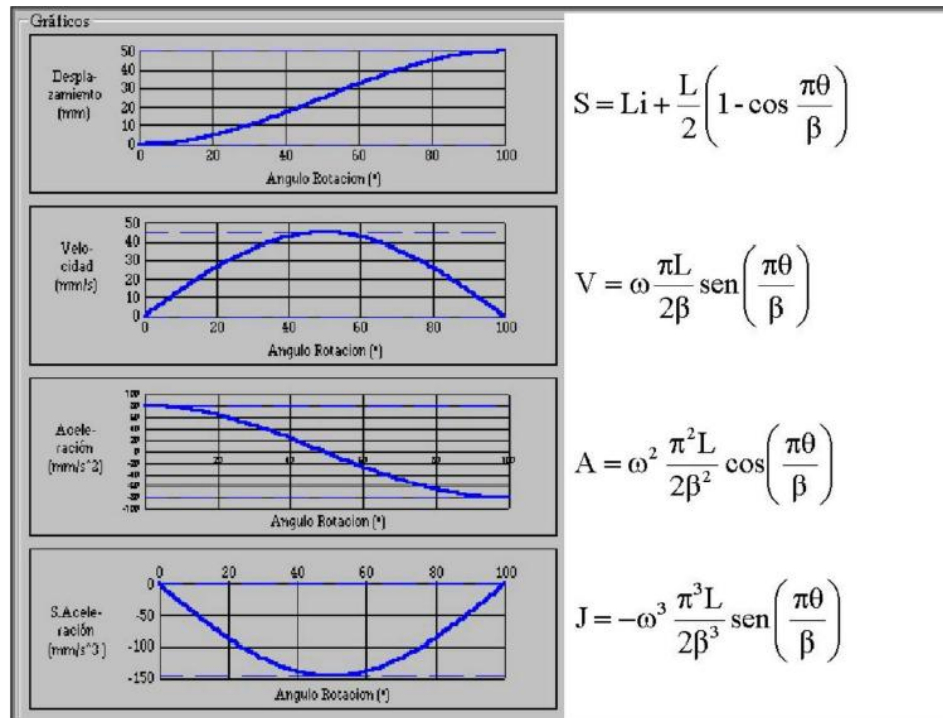


Figura 22. Movimiento armónico simple de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.

Cabe destacar que en programas de ascenso y descenso con este tipo de curvas, donde ambos tramos tienen 180° , la aceleración se vuelve continua. Esta combinación es más conocida como leva excéntrica, que no es otra que una leva con forma circular girando alrededor de un eje que no pasa por el centro del círculo. Hay que decir que eso sólo ocurre para seguidores de cara plana traslacionales ya que los otros tipos de seguidores dan lugar a formas de leva que no son circulares.



Movimiento Cicloidal

Viendo el caso anterior, nos hace pensar que el diseñador no sólo debe considerar la función de desplazamiento para el diseño de la leva, debe ir más allá, a sus derivadas.

Según las referencias estudiadas del profesor Norton, la mejor consideración es estudiar primero las derivadas superiores, en especial la aceleración. Según el profesor, la función de aceleración, y en menor grado la función de sobreaceleración, deberán ser de primordial interés para el diseñador.

El movimiento cicloidal, a diferencia del anterior, no muestra incompatibilidades en su curva de aceleración. La razón es que su deducción parte de una aceleración senoidal de periodo completo, y por tanto con valores cero en sus extremos, lo que permite combinación con detenimientos.

Sin embargo, las curvas cicloidales no son una buena elección cuando se desean movimientos de ascenso y descenso consecutivos consigo misma, pues da lugar a valores pico muy elevados en velocidad y aceleración, además de discontinuidades en sobreaceleración.

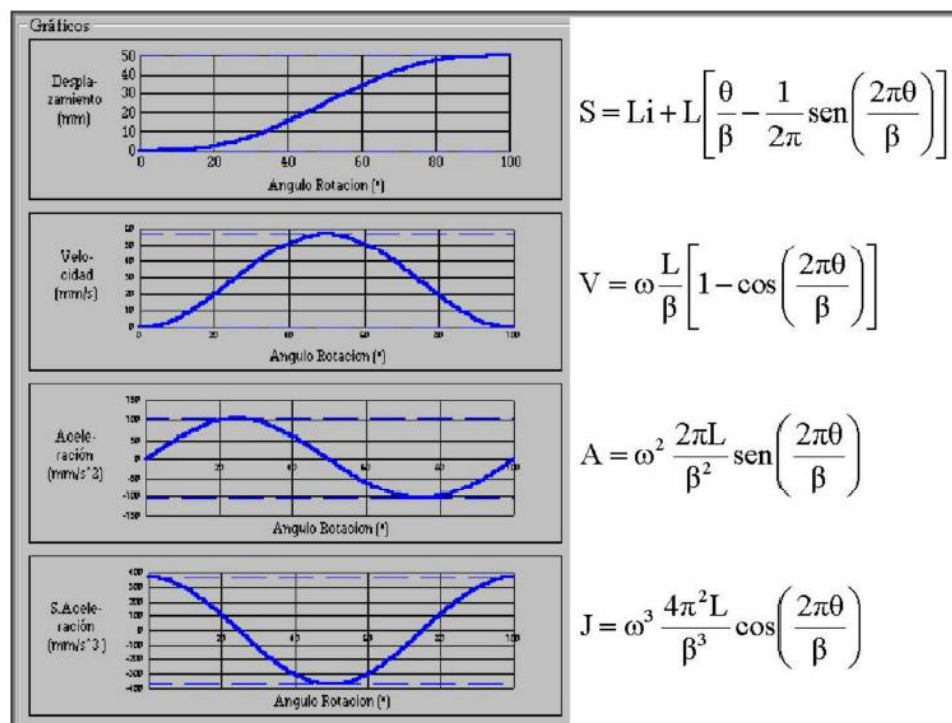


Figura 23. Movimiento cicloidal de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $Li=0$ mm.



Polinómico 3-4-5

La clase de funciones polinomiales es uno de los tipos más versátiles que puede utilizarse en el diseño de levas. No se limitan a aplicaciones concretas y pueden adaptarse a muchas especificaciones de diseño. La forma general de una función polinomial es:

$$s = C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5 + C_6 x^6 + \dots + C_n x^n$$

Donde s es el desplazamiento del seguidor, x es la variable independiente, que en este caso será reemplazada por θ/β o el tiempo t . Los coeficientes C_n son las incógnitas a determinar en el desarrollo de la ecuación particular que satisfaga una especificación de diseño.

El polinomio más sencillo de esta familia que nos podemos encontrar es el comúnmente denominado 3-4-5. Esto se debe porque al aplicar condiciones de contorno nulas en la primera y segunda derivada, en ambos extremos del intervalo, los coeficientes C_0 , C_1 y C_2 se hacen cero.

En este caso la función de sobreaceleración da lugar a discontinuidades en la combinación con otras curvas porque no se impuso ninguna restricción en los valores de frontera de dicha función.

Este polinomio es combinable con detenimientos o cualquier otro tipo de movimiento con velocidad y aceleración cero en su punto de enlace.

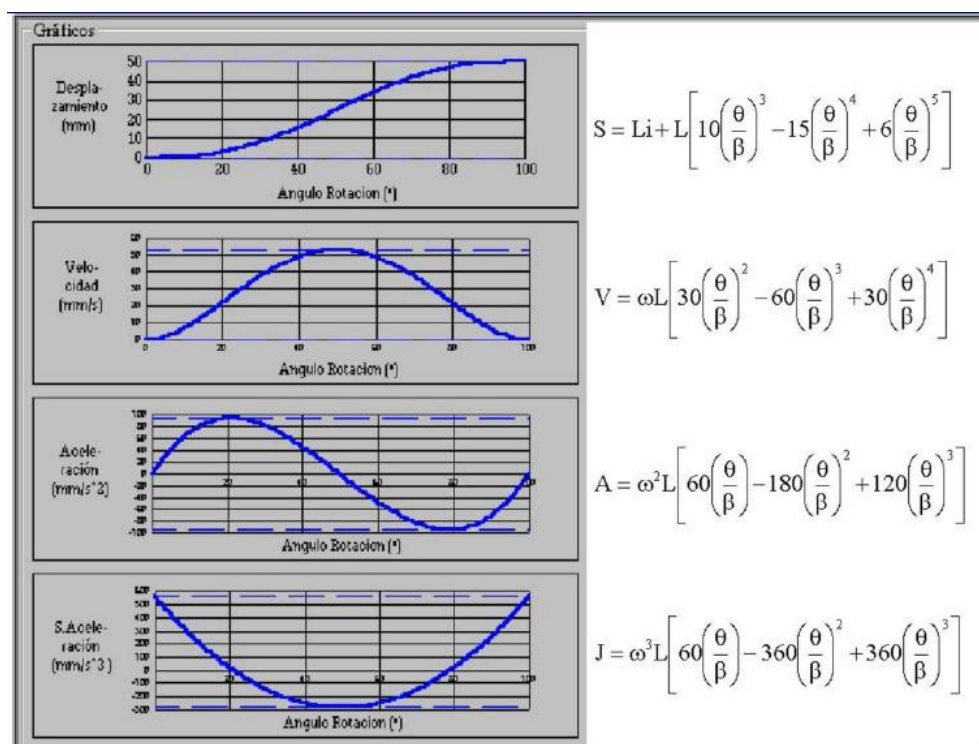


Figura 24. Movimiento polinómico 3-4-5 de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $Li=0$ mm.



Polinómico 4-5-6-7

Con esta curva se trata de remediar las posibles discontinuidades que puede llevar a dar la función de sobreaceleración en las curvas polinómicas 3-4-5, limitando a cero ambos extremos en la subida. Para ello se ha de aumentar el grado del polinomio para que pueda contener dos condiciones de contorno adicionales.

De nuevo este polinomio puede combinarse junto con detenimientos o con cualquier movimiento que tenga condiciones nulas de velocidad, aceleración y sobreaceleración en su punto de unión con el tramo polinómico.

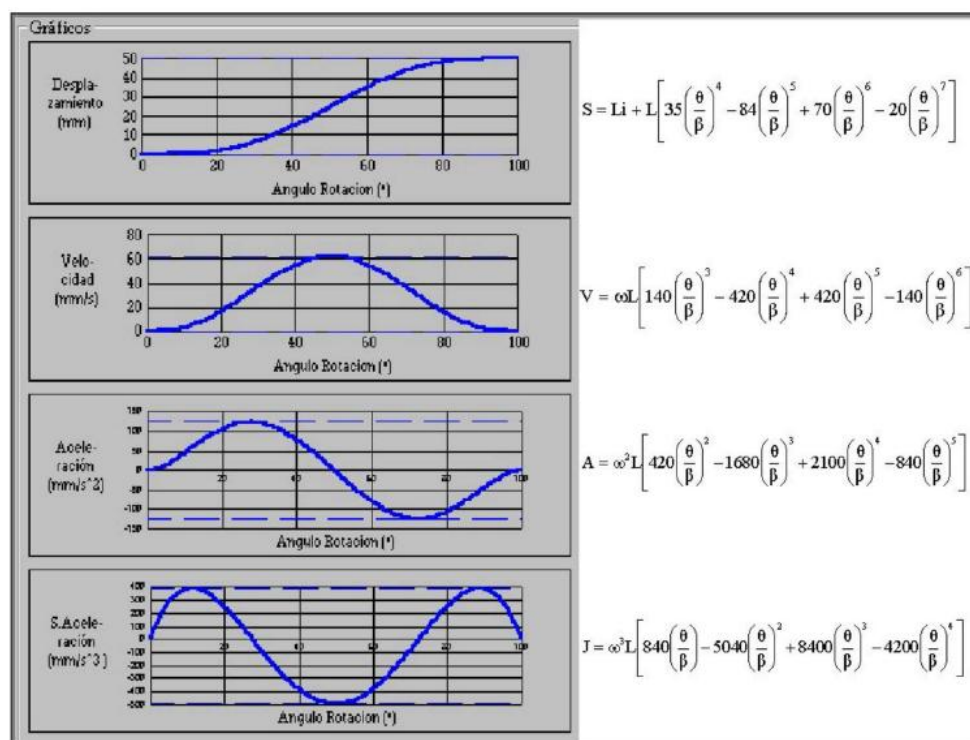


Figura 25. Movimiento polinómico 4-5-6-7 de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $Li=0$ mm.

Velocidad constante

Un tramo de velocidad constante consiste en un desplazamiento lineal. La posición inicial viene dada por la ordenada del último tramo, que será cero en caso de que sea el primero. Por otro lado, la posición final se determina por la magnitud del desplazamiento, previamente configurado por el diseñador, y su sentido.

Este tipo de movimiento no es fácilmente conjugable con otros movimientos. Tampoco resulta combinable con detenimientos, donde la velocidad es nula. Para grandes velocidades de giro resulta un tipo de movimiento poco adecuado por el pico que produce sobre el palpador en su desplazamiento. Este tipo de tramo suele emplearse junto con movimientos de medio periodo.



Movimientos de medio periodo

Este tipo de movimientos permiten la combinación con movimientos de velocidad constante ya que tienen velocidad no nula en sus extremos, al contrario que los armónicos simples y cicloidales.

Como su nombre indica, se trata de curvas de medio periodo con velocidad no nula en uno de sus extremos. De esta manera se tienen dos semiarmónicas y dos semicicloidales, ya que la condición de velocidad distinta de cero puede estar al principio o al final del intervalo.

La diferencia más notable entre ambos es que el movimiento semicicloidal tiene aceleración nula al comienzo y al final del tramo, no como el movimiento semiarmónico. Como hemos visto, estas diferencias se han dado en los movimientos de periodo completo.

- Semiarmónico de velocidad inicial nula ($V_0 = 0$)

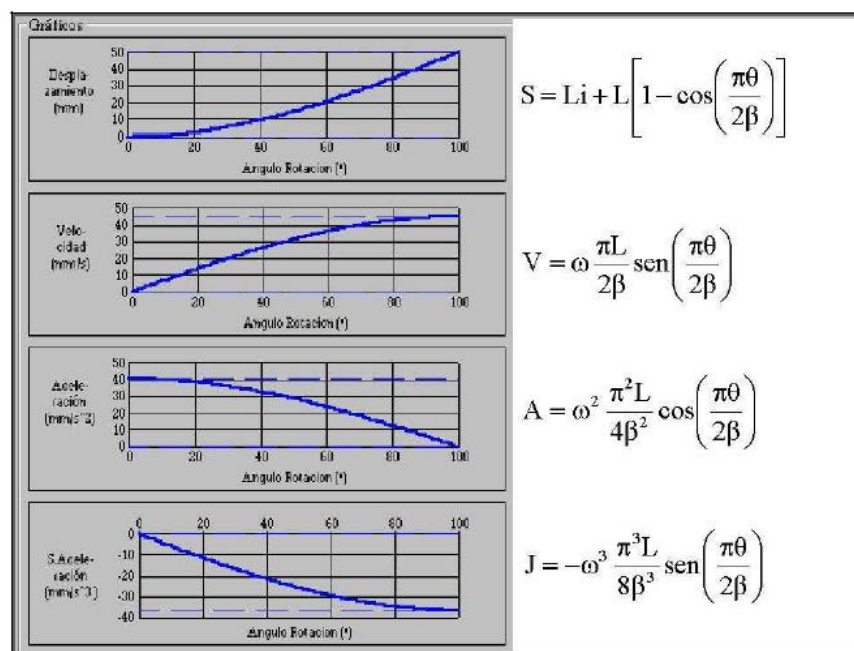


Figura 26. Movimiento semiarmónico de ascenso de V_0 nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.



- Semiarmónico de velocidad final nula ($V_f = 0$)

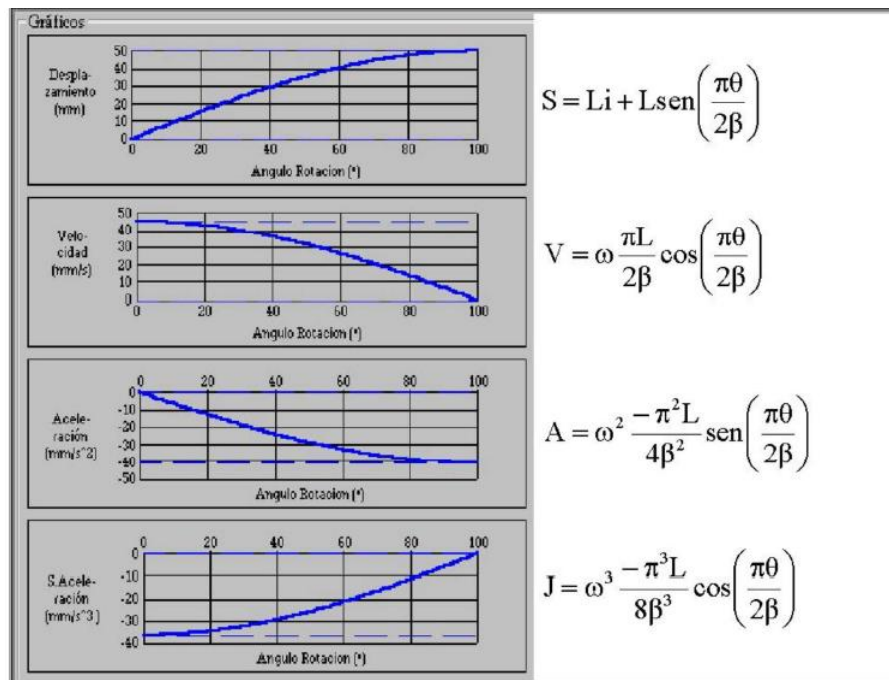


Figura 27. Movimiento semiarmónico de ascenso de V_f nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.

- Semicicloidal de velocidad inicial nula ($V_0 = 0$)

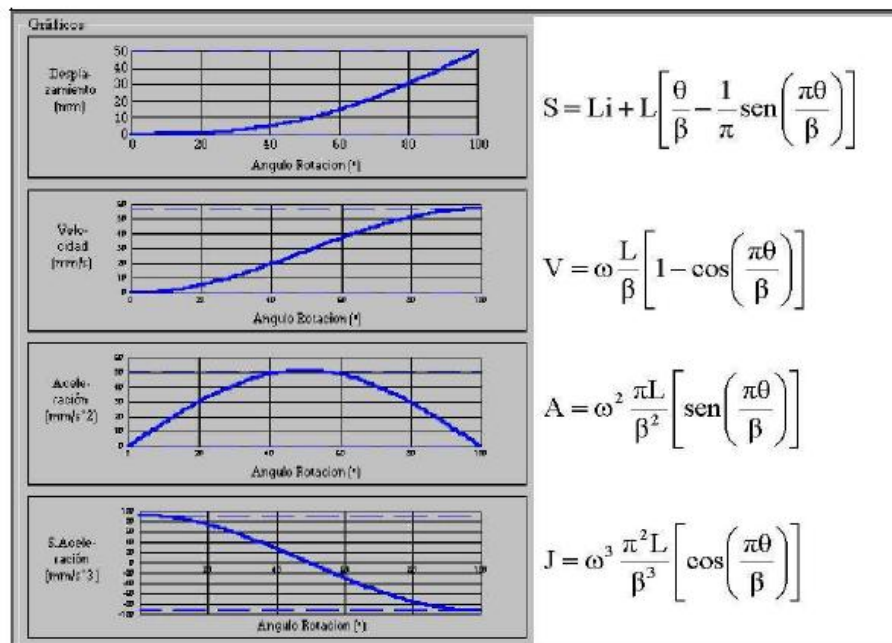


Figura 28. Movimiento semicicloidal de ascenso de V_0 nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.



- Semicicloidal de velocidad final nula ($V_f = 0$)

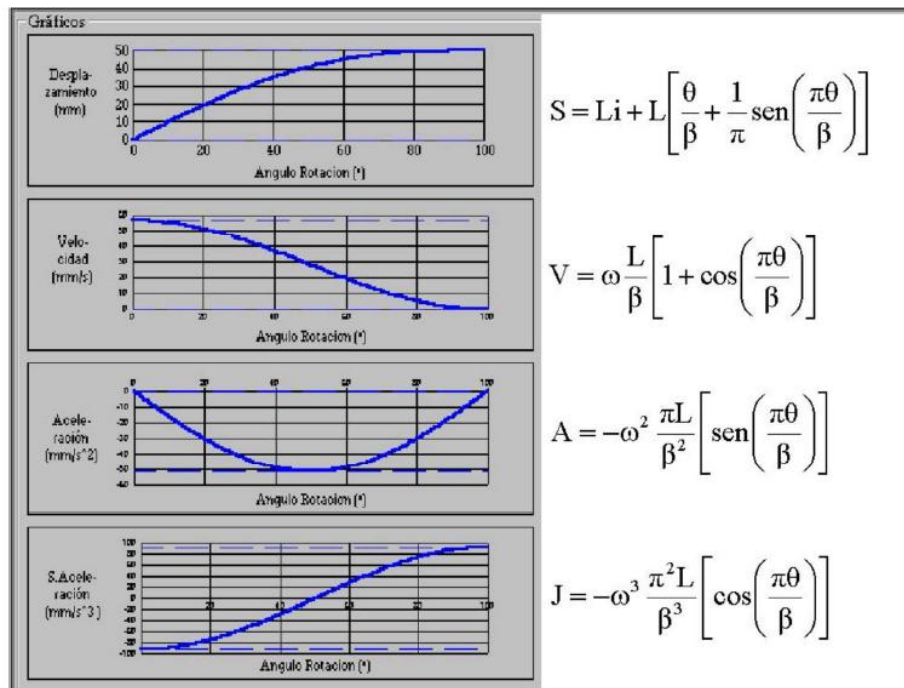


Figura 29. Movimiento semicicloidal de ascenso de V_f nula. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.



Movimiento Armónico Doble

Este es un movimiento de ascenso y descenso creado para evitar que la aceleración retorne a cero cuando finaliza la subida y va a comenzar la bajada. De esta manera, el descenso comienza con el mismo valor de aceleración en el que terminó la subida.

Ambos tramos, el de ascenso y descenso, deben tener la misma β para que se cumpla la ley fundamental de la continuidad.

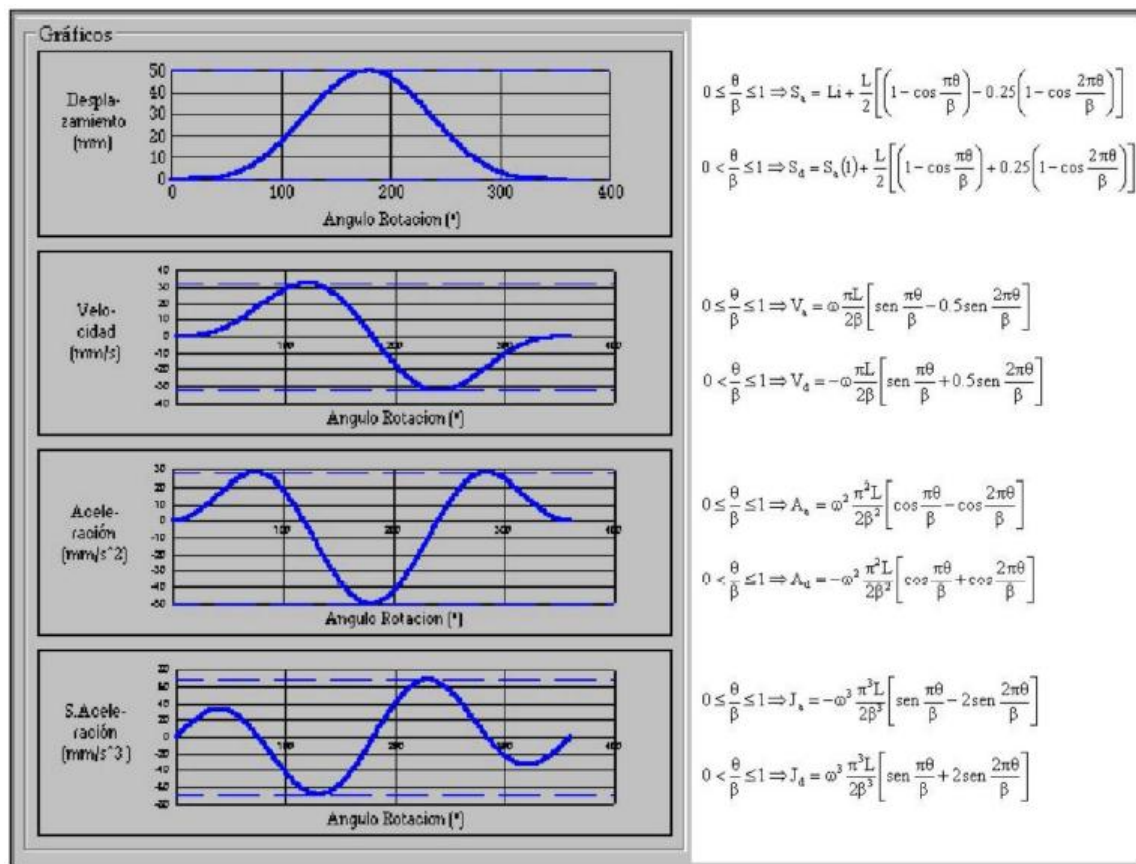


Figura 30. Movimiento armónico doble de ascenso. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.



Movimientos compuestos

A diferencia de las curvas anteriores, en este apartado describiremos algunas cuyas expresiones se componen a su vez por tramos, es decir, no son una sola función en sí. De esta forma se trata de reunir las mejores características de otros movimientos simples en función de la porción de tramo para la que se requieran.

- Aceleración de Onda Senoidal Modificada: su diagrama de aceleración resulta de la combinación de ondas senoidales de distinta frecuencia (partiendo de un movimiento cicloidal). De esta manera es posible retener alguna característica de la uniformidad de la curva cicloidal, reduciendo su aceleración pico.

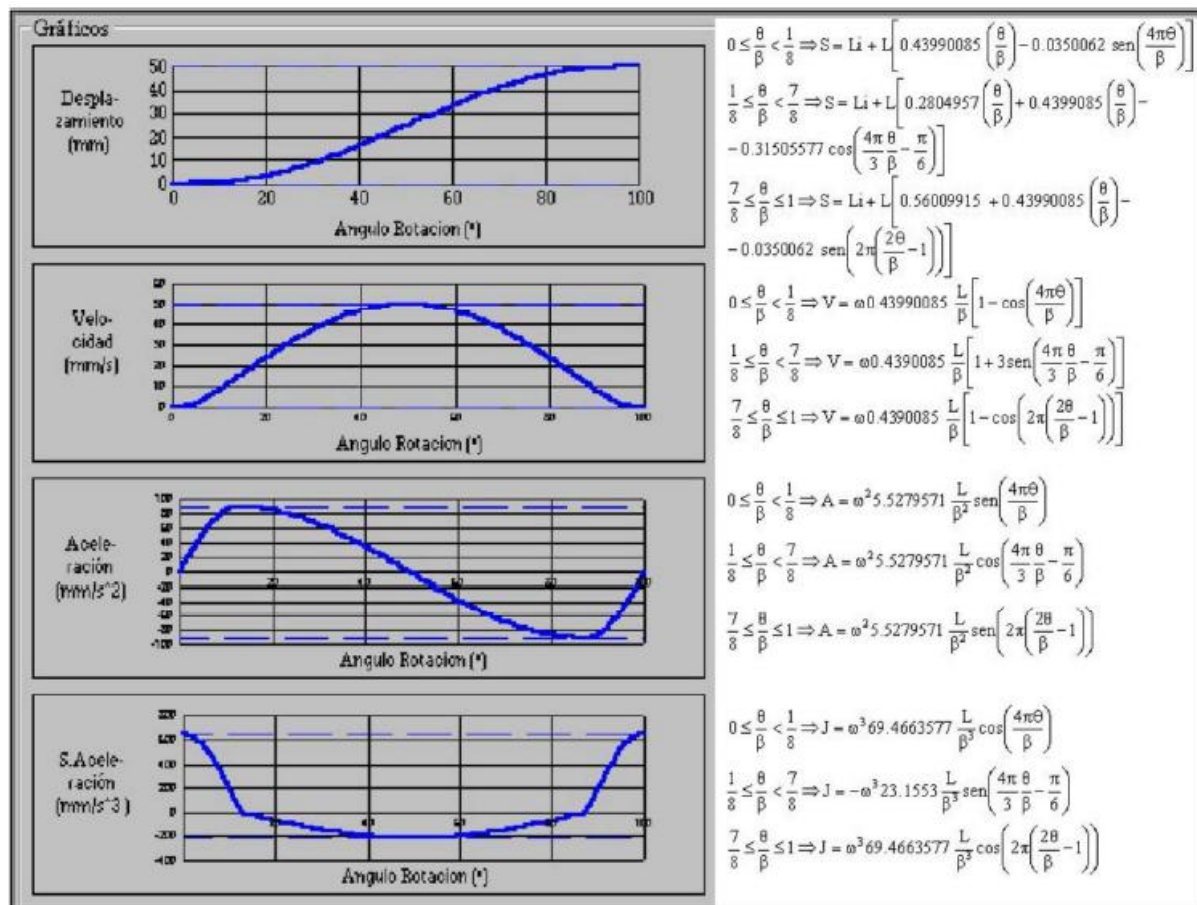


Figura 31. Movimiento de ascenso de aceleración de onda senoidal modificada. $L=50$ mm, $\beta=100^\circ$, $\omega=1$ rad/s, $Li=0$ mm.

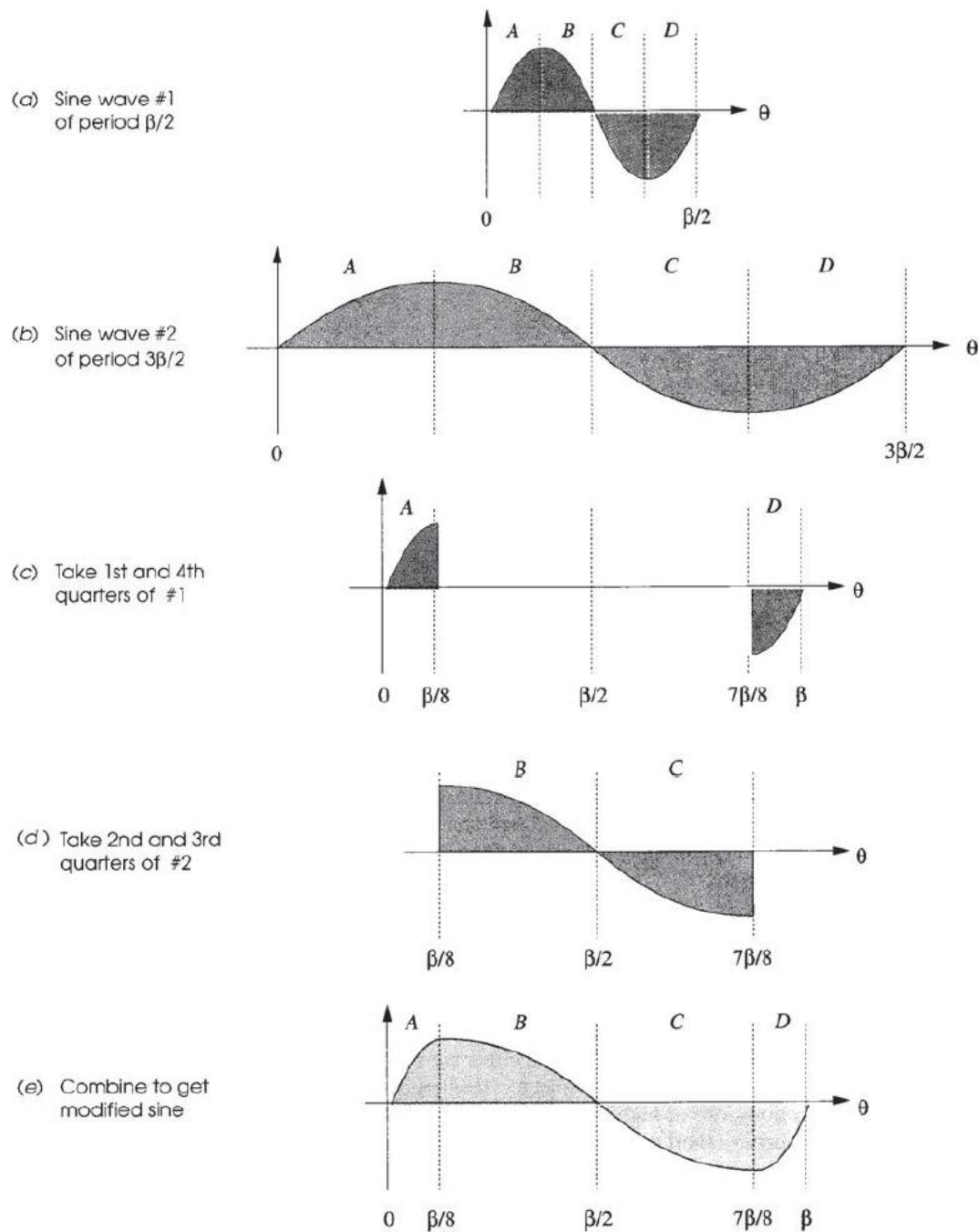


Figura 32. Construcción de la función de aceleración senoidal modificada. [Norton, 2009].



- Aceleración de Onda Trapezoidal Modificada: se trata de una curva de aceleración de onda cuadrada que ha sido modificada para evitar discontinuidades en puntos extremos y medio si se combinase con detenimientos. Conceptualmente consiste en sustituir las esquinas de las ondas cuadradas por partes de funciones seno para crear una transición suave y evitar discontinuidades. Popularmente utilizada para levas de doble detenimiento.

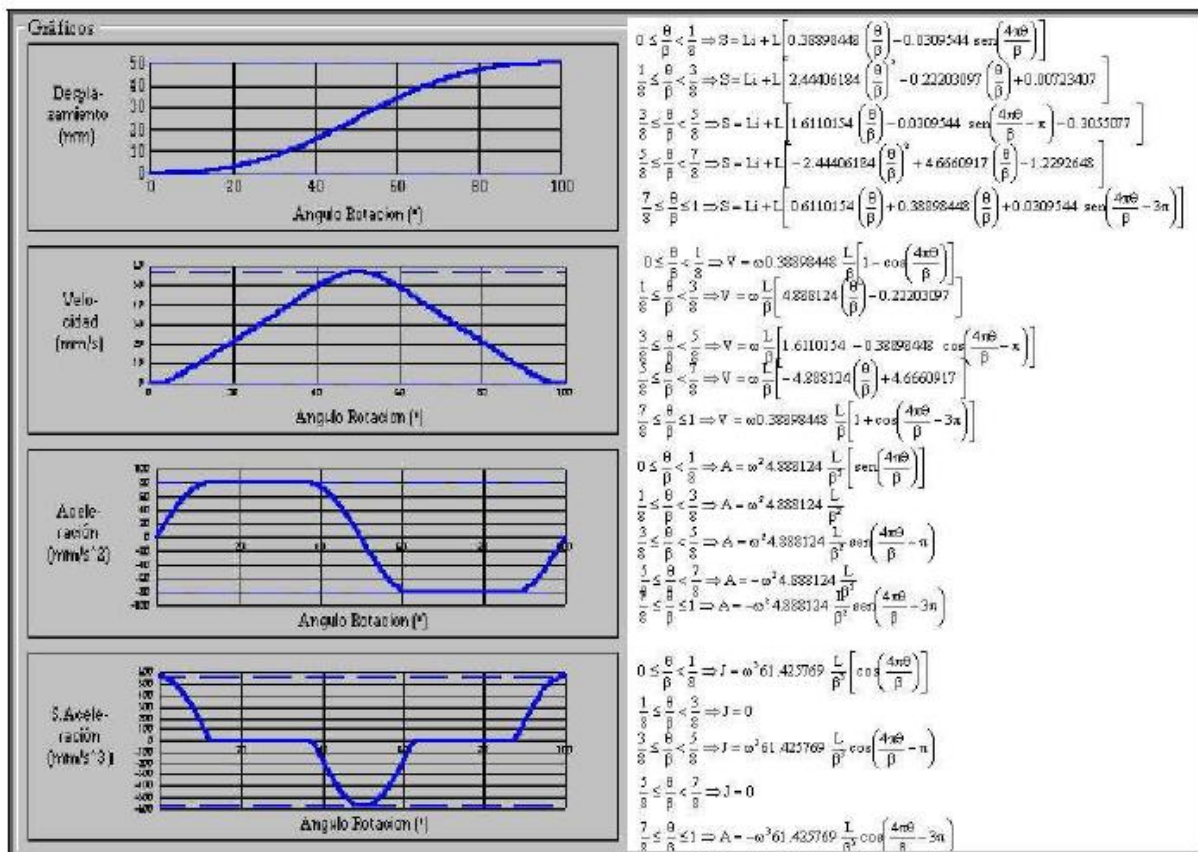


Figura 33. Movimiento de ascenso de aceleración de onda trapezoidal modificada.
L=50 mm, $\beta=100^\circ$, $\omega=1$ rad/s, $L_i=0$ mm.

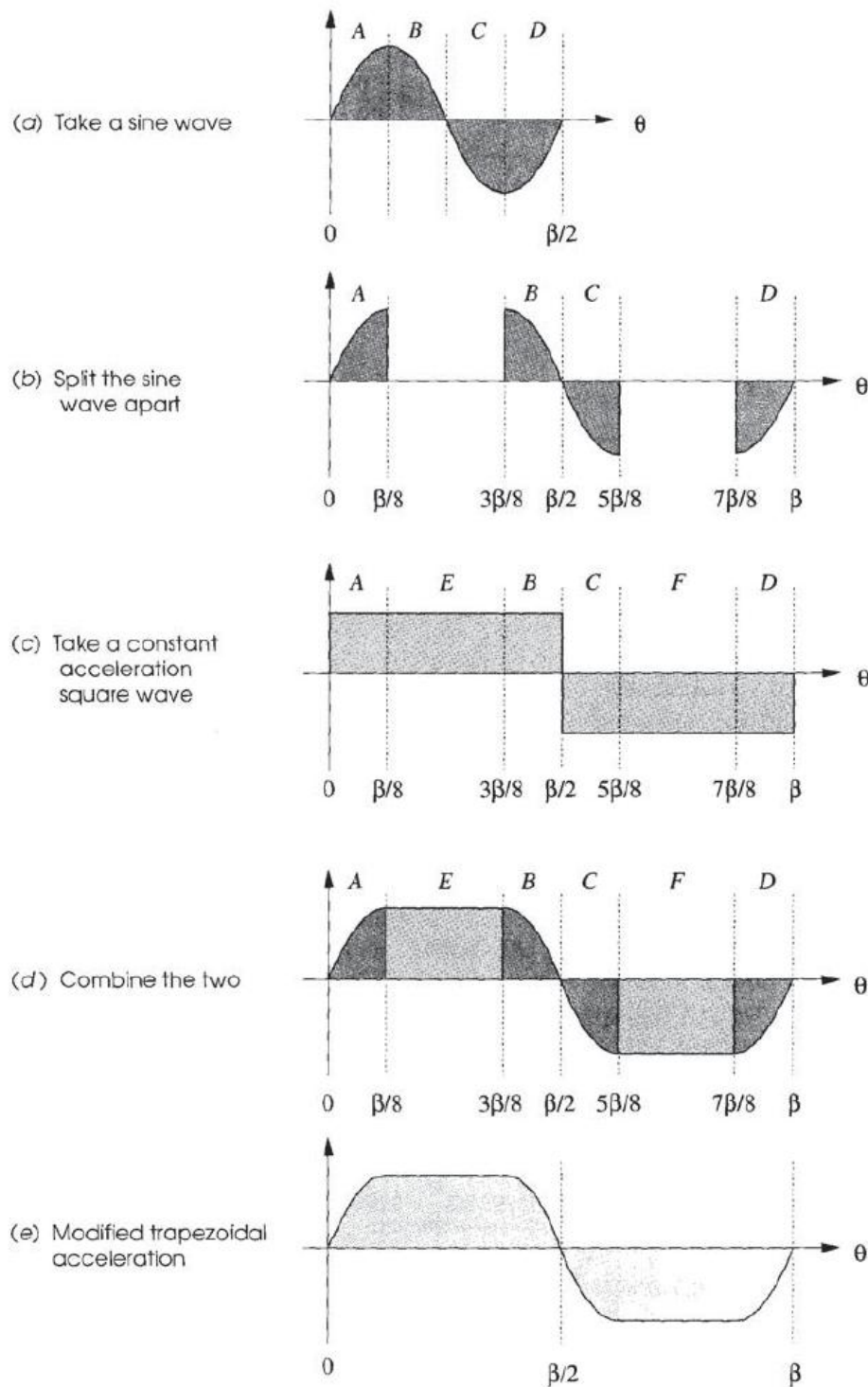


Figura 34. Construcción de la función de aceleración trapezoidal modificada. [Norton, 2009].



2.7. DIMENSIONAMIENTO DE LA LEVA

Tras definir las funciones S , V , A y J , el diseñador debe proceder al dimensionamiento de la leva. En este apartado se considerará que en el paso de definición de diagramas se cumplen perfectamente las leyes fundamentales del diseño de levas, principalmente la continuidad. Esto es realmente sencillo cuando se conoce profundamente el efecto que tiene cada función matemática sobre los diagramas cinemáticos, y de qué manera se pueden combinar estas curvas para obtener los diagramas dentro de un marco que nos permita avanzar hasta el mecanismo final.

En este apartado vamos a comentar dos factores fundamentales que condicionarán el tamaño de la leva: el ángulo de presión y el radio de curvatura. Además de estos parámetros que analizaremos con detalle, existen otras características de los mecanismos leva-seguidor que conviene optimizar para que el sistema funcione de manera eficiente. Con esto nos podemos referir a la excentricidad del seguidor que muestra con el centro de giro de la leva, o las dimensiones de las guías y de los palpadores de traslación.

Ángulo de presión

El ángulo de presión Φ determina la relación entre la fuerza de contacto F leva-palpador y la componente de ésta en la dirección del movimiento del palpador como podemos ver en la Figura 35. Cuando dicho ángulo es 0° , toda la fuerza de transmisión se convierte en movimiento del seguidor y ninguna en velocidad de deslizamiento. Cuando llega a 90° , el seguidor está completamente retenido.

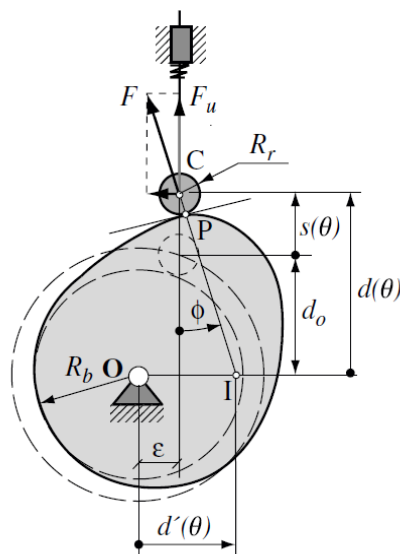


Figura 35. Ángulo de presión en leva con palpador de rodillo.



En la práctica se recomienda que el máximo ángulo de presión absoluto esté entre 0 y unos 30° para palpadores de traslación, con el fin de evitar el acúñamiento del palpador en sus guías y minorar las excesivas cargas laterales; y hasta unos 35° para palpadores oscilantes o rotacionales.

Unos valores superiores a los comentados podrían incrementar la fricción en el pivote o deslizante del seguidor a niveles indeseables, tendiendo a trabar al seguidor en sus guías. El ángulo de presión es más crítico en el tramo de subida que de bajada, ya que en la subida el movimiento del palpador es contrario a las cargas externas, y la fuerza útil debe ser mayor que las fuerzas resistivas, lo cual sólo será posible para valores pequeños de Φ .

De la propia definición dada del ángulo de presión se puede deducir que para un seguidor de cara plana, ya sea traslacional u oscilante, si la dirección de aplicación de la fuerza es normal a la superficie de contacto, entonces será perpendicular a su cara y coincidirá con la dirección de su movimiento, por lo que el ángulo de presión será nulo para este tipo de seguidores. Visto de otra manera, la dirección de aplicación de la fuerza y la dirección de su movimiento son iguales. Es por ello que para estos seguidores no consideraremos Φ como un parámetro a tener en cuenta en los cálculos.

Aun así, siempre pueden surgir excepciones. En el caso en el que la cara plana no fuese perpendicular a su movimiento podría aparecer un ángulo de presión. En este caso, el ángulo de presión coincidiría con la inclinación de la cara siendo $\beta = \Phi$ como vemos en la Figura 36. En estos casos no vamos a entrar en detalle ya que no los contemplaremos a la hora de implementar la aplicación.

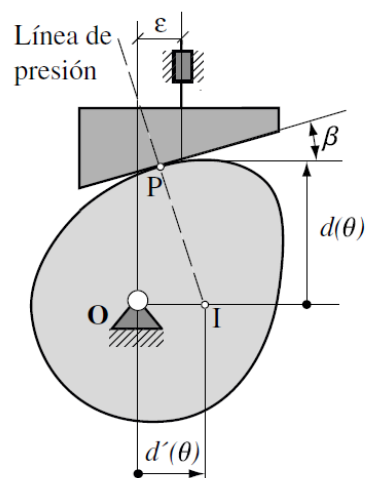


Figura 36. Leva con palpador plano de cara inclinada.



El caso que nos interesa a nosotros es el cálculo de este ángulo para palpadores circulares de traslación, que obtendremos a partir del análisis de la Figura 35. En estos seguidores, utilizaremos como punto de cálculo su centro, desde donde consideraremos el movimiento. En este tipo de seguidores, el ángulo de presión ofrece una idea de la facilidad con la que la leva transmite el movimiento al seguidor. Cuando el valor es muy elevado, la posibilidad de atasque es mayor y el seguidor comenzará a moverse con dificultad.

Entonces, desde la figura correspondiente podemos observar que:

$$\Phi = \arctan \left[\frac{d'(\theta) - \varepsilon}{d(\theta)} \right]$$

Donde:

$$d(\theta) = d_0 + s(\theta), \quad d_0 = \sqrt{(R_b + R_r)^2 - \varepsilon^2}$$

Del análisis de la expresión anterior se observa que el radio del círculo base R_b está relacionado con el ángulo de presión Φ . Al aumentar R_b disminuye el valor de Φ , pero aumentan las dimensiones de la leva. Para corregir un ángulo de presión elevado, otra opción sería aumentar el radio del rodillo o modificar la excentricidad. Este último parámetro, si el diseñador es capaz de seleccionarlo adecuadamente, permite reducir las dimensiones de la leva sin afectar a la eficacia del mecanismo, manteniendo el ángulo de presión dentro de unos valores aceptables.

Un valor positivo de la excentricidad hará disminuir el ángulo de presión en el tramo de subida, donde $d'(\theta) > 0$, pero lo aumentará en el tramo de bajada. A su vez, la excentricidad negativa hará lo contrario. Es por ello que el diseñador debe encontrar un compromiso entre este parámetro y los radios base y del rodillo para optimizar lo mejor que pueda su sistema, encontrando unas tolerancias adecuadas para el ángulo de presión, según el espacio que vaya a ocupar su mecanismo dentro de una posible máquina.

En el caso de los seguidores circulares oscilantes conforme aumenta el ángulo de presión también lo hace el rozamiento en el pivote sobre el que gira el brazo. Hay que tratar de reducir estos efectos tan perjudiciales para el buen funcionamiento del mecanismo, lo cual puede llegar a ser más complicado que en el caso anterior, porque hay que tener en cuenta varios parámetros más.

Al aumentar el radio del seguidor ocurre algo semejante a lo que pasaba con la excentricidad en el caso del rodillo traslacional: la curva del ángulo de presión resultante queda por debajo. Si se hace más pequeño se daría el efecto contrario, de modo que hay que buscar un valor intermedio del radio del seguidor para tratar de minimizar el máximo del ángulo de presión en valor absoluto. Tendremos problemáticas similares cuando variemos otras



magnitudes, como el radio base, la longitud del brazo del seguidor, o la distancia entre éste y el centro de giro de la leva.

Se aprecia que puede resultar verdaderamente difícil llegar a una solución suficientemente buena. Son demasiadas magnitudes a considerar, y por si fuese poco, el ángulo de presión no es el único criterio a tener en cuenta en el cálculo, también tenemos que considerar el radio de curvatura del perfil de la leva.

A pesar de los inconvenientes y quebraderos de cabeza que puede resultar la optimización del ángulo de presión en este tipo de seguidores, de forma general suelen mostrar una mejora sobre los seguidores trasladantes en la distribución de fuerzas, permitiendo un mayor margen en el ángulo de presión. De esta forma, se pueden reducir más los tamaños de estas levas, encontrándolas en multitud de aplicaciones. Basta con un diseño apropiado y no uno más optimizado para que el seguidor no se retenga impidiendo el movimiento de la leva.

Como conclusión, podemos afirmar que el proceso de optimización del ángulo de presión resulta más cómodo para levas con seguidor de rodillo traslacional, sin embargo, cuando sólo nos interesa cubrir una aplicación sin demasiadas restricciones o limitaciones de contorno, una leva con seguidor de rodillo oscilante trabajará mejor y permitirá un abanico más amplio de posibilidades antes de llegar a un límite de retención.

Radio de curvatura

El radio de curvatura es una propiedad matemática de una función. Su valor no está limitado a levas, pero es importante. El radio de curvatura representa numéricamente la mayor o menor concavidad/convexidad que muestra la representación gráfica de una función. Así por ejemplo, una recta tiene radio de curvatura infinito y una circunferencia un radio de curvatura finito.

En sí, el concepto es realmente simple. No importa lo complicada que pueda ser la forma de una curva, ni como de alto sea el grado de la función que la describe, pues tendrá un radio de curvatura instantáneo en cada punto de la curva. A su vez, estos radios de curvatura tendrán unos centros instantáneos.

Para entrar en detalle en este concepto nos basaremos en el libro referenciado [1] del profesor Norton. En dicho libro se nos explica el concepto de rebaje como la problemática que surge cuando el radio del seguidor de rodillo es mayor que el radio local más pequeño positivo en la leva, es decir, el radio de curvatura en una posición determinada, que viene dado por el radio de la curva de paso en dicho punto.

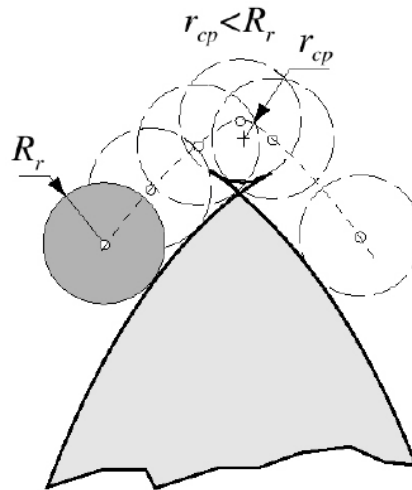


Figura 37. Fenómeno de rebaje.

En la figura vemos como a pesar de que la ley de desplazamientos sea continua y suave se pueden dar situaciones de autointersecciones, porque matemáticamente es posible. En este caso, se impediría seguir con la ley de desplazamientos especificada, ya que aparecen unos movimientos virtuales que no podemos admitir. La leva no sería factible.

Por otro lado, R.L. Norton nos define el concepto de cúspide, que sería cuando el radio del seguidor está en un punto exactamente igual al radio de curvatura mínimo convexo de la leva. Es decir, que en el proceso de fresado de esta leva se generaría una punta perfecta, o cúspide, lo que es completamente inviable a altas velocidades. Esta leva ya no tendría la misma función de desplazamiento original.

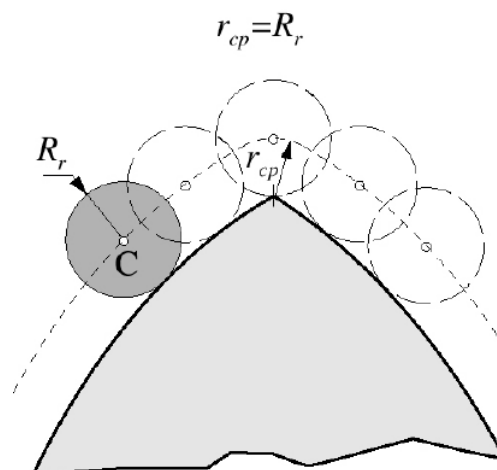


Figura 38. Fenómeno de cúspide.



Esta situación no es deseable para ningún tipo de seguidor. Siempre que se pueda evitar habrá de hacerse, pero bien es cierto que en el caso de un seguidor de cara plana, dicho palpador podrá acceder al punto teórico sin mayor problema, a pesar de que se generarían presiones muy grandes en el punto de contacto.

Visto esto, lo que ahora hace falta decidir es qué valores de radio de curvatura son indicativos de la presencia de cúspides o rebajes en el perfil de la leva, con el fin de evitar que se den tales situaciones. Nos vamos a centrar en el caso de levas planas de rotación con seguidores traslacionales de rodillo.

Lo primero es definir el radio de curvatura de la curva de paso, ρ_{paso} , para este tipo de seguidores:

$$\rho_{\text{paso}} = \frac{[(R_p + s(\theta))^2 + (s'(\theta))^2]^{3/2}}{(R_p + s(\theta))^2 + 2 * (s'(\theta))^2 - (R_p + s(\theta)) * s''(\theta)}$$

Dónde:

$$R_p = R_b + R_r$$

Como sabemos que el radio de curvatura del perfil de la leva, ρ , y el de la curva de paso, ρ_{paso} , difieren en el radio del rodillo, R_r , podemos plantear la siguiente ecuación:

$$\rho = \rho_{\text{paso}} - R_r$$

Así, asumiremos los siguientes casos:

- Cuando $\rho < 0$, hay tramos cóncavos en el perfil de la leva. Se ha de verificar que $|\rho| > R_r$. En caso contrario, el rodillo no puede acceder al punto teórico de contacto leva-seguidor.
- Si $\rho_{\text{paso}} > 0$, hay tramos convexos de la curva de paso. Se debe de verificar que $\rho_{\text{paso}} > R_r$. En el caso de $\rho_{\text{paso}} = R_r$, se generaría un vértice ($r=0$) en el perfil de la leva. Si $\rho_{\text{paso}} < R_r$, ocurriría el fenómeno del rebaje.

Para evitar situaciones conflictivas, debemos conseguir que $R_r < \rho_{\text{paso}}$. Para ello podríamos disminuir la excentricidad, aumentando así el radio de curvatura, o evitar un rebaje disminuyendo R_r y/o aumentando R_b .

En el caso de utilizar un palpador de cara plana, como hemos visto, la situación más perjudicial no sería la aparición de cúspides sino de rebajes, ya que además de originar un pico, este fenómeno impediría seguir la ley de desplazamiento especificada porque el perfil se cruza sobre sí mismo.



2.8. CONSIDERACIONES PRÁCTICAS DE DISEÑO

En la presente sección de este segundo capítulo se busca dar unos ápices en el diseño de levas que sirvan al diseñador como breves pautas en la confección de estos sistemas. Nos basaremos en la experiencia de más de 40 años como diseñador de levas y maquinaria de nuestro autor de referencia, el profesor Robert L. Norton, quien trata de resolver situaciones que puedan ser desconcertantes en la fase del diseño de levas [2].

Conviene entender a fondo la tipología de estos elementos para tomar decisiones correctas y de forma temprana en su diseño, pues de lo contrario, decisiones arbitrarias pueden traducirse, a la larga, en consecuencias importantes y costosas. Por supuesto, la información presentada a continuación siempre puede ser extendida y pueden darse casos y casos donde convenga dirigirse por otro camino. Se tratará de dar unas pequeñas pinceladas en los aspectos de diseño más controvertidos que se puede encontrar, sin profundizar en exceso en la materia.

¿Seguidor traslacional o rotacional?

La elección de uno u otro tipo de seguidor viene determinada la enorme mayoría de veces por requisitos de diseño y funcionales de la máquina final. Pero en el caso de disponer de cierta libertad para elegir entre uno u otro, el profesor Norton nos recomienda el uso de seguidores rotacionales.

Para movimientos de salida oscilantes, la elección es clara. Pero es que para mecanismos donde se busquen como resultado movimientos lineales, resulta más eficiente el uso de sistemas de leva con seguidor rotacional que proporcionen un movimiento de línea recta aproximado, el cual es perfectamente factible con un seguidor rotacional de radio grande.

En el caso de este tipo de mecanismos, el diseñador se asegura que el seguidor va a estar siempre alineado en el mismo plano de la leva sin necesidad de utilizar guías auxiliares más que su propio pivote. Un ejemplo muy común donde se utilizan levas rotacionales con seguidor oscilante son los mecanismos de apertura y cierre de las válvulas de admisión de motores de combustión interna, donde desde el árbol de levas se empuja al seguidor, que a través de un balancín transmite un movimiento lineal a estas válvulas.

¿Cierre de forma o de fuerza?

Ante esta pregunta cabe destacar que las levas con cierre de forma (vía o ranura) siempre son más caras y costosas en su fabricación que las levas con cierre de fuerza o abierta. Ante las dificultades de mecanizado de las ranuras en la leva que sirven de guía para el seguidor, se suma el complejo del fenómeno térmico.

Durante su funcionamiento, el efecto de las altas temperaturas debido al rozamiento de los elementos del sistema puede provocar distorsiones en el guiado de seguidor, perdiendo el ajuste rápidamente y obligando a un mantenimiento periódico de rectificado. Las levas con



sistema de cierre de fuerza también sufrirán leves variaciones en la geometría debido a estos efectos térmicos, pero la vigilancia del mantenimiento es menos exhaustiva puesto que el seguidor no debe ajustar perfectamente en ningún riel de guiado.

Por otro lado, la principal ventaja de una leva cerrada por su forma es que evita el uso de muelle o resorte de retorno como se utilizaría en un sistema de cierre por fuerza. En estos últimos sistemas de cierre, velocidades altas de giro pueden provocar enormes vibraciones en el seguidor con unas consecuencias nefastas, además de considerar una posible resonancia entre la masa del resorte y del seguidor, desde un aspecto más dinámico. Estas altas velocidades de la leva también pueden desatar la separación o despegue del seguidor, llevando a la necesidad de utilizar resortes más rígidos.

Pero también se pueden encontrar ventajas en este resorte de retorno, y es su capacidad para absorber los cambios de signo de la aceleración del seguidor durante su movimiento. En el caso de levas con cierre de forma, no tenemos ningún elemento que sea capaz de minimizar estos esfuerzos dinámicos, lo que provoca ruido y desgastes locales. Este fenómeno de cambio de signo de la aceleración es lo que R. L. Norton denomina “choque por cruce”. Diversos estudios han demostrado que los seguidores de rodillo que funcionan en contacto con una leva radial abierta bien lubricada tienen tasas de deslizamiento de menos del 1% [1].

Por lo tanto, el uso de una u otra configuración de cierre de los sistemas leva-seguidor vendrá muchas veces determinada por las velocidades de giro a las que se va a someter nuestra leva. Hemos visto que ante velocidades altas, los resortes son vulnerables al fallo con la consecuencia del despegue del seguidor, algo que queremos evitar. Pero con un muelle de rigidez adecuada podemos absorber los efectos dinámicos de la aceleración, teniendo un diseño compacto y seguro.

¿Seguidor de rodillo o de cara plana?

En un principio diríamos que la mejor opción desde el punto de vista del diseño de la leva es utilizar un seguidor de rodillo, pues acepta radio de curvatura negativos con alta seguridad ya que suele cumplir las condiciones de radios de curvatura expuestos en la sección anterior, permitiendo por lo tanto gran variedad en el programa de movimiento de la leva.

Además de esto, son más fáciles de producir y de encontrar en el mercado. Su estandarización es enorme, y la capacidad para seleccionar y personalizar un mecanismo de leva-seguidor con este tipo de palpadores es muy alta.

Dentro de este tipo de seguidores, se puede realizar una distinción entre seguidores cilíndricos y esféricos. Los primeros, siempre que en su colocación estén bien alineados en el plano que contiene a la leva, producirán menos tensiones y serán más eficientes que los esféricos. Por otro lado, los de forma esférica permiten un pequeño margen en su colocación



sin perder precisión en el movimiento de salida. Se debe tomar precaución a la hora de utilizar estos últimos, pues si ruedan por el borde de la leva causarían un rápido desgaste.

Por otro lado, los usuarios más importantes de levas con seguidor plano son los fabricantes de motores de combustión interna. Su producción es lo suficientemente alta como para que sea económico solicitar este tipo de palpadores en grandes volúmenes, incluso más que los palpadores de rodillo. Una gran ventaja que tienen estos seguidores es su capacidad para ahorrar espacio, ajustándose perfectamente al volumen permitido en la máquina.

Cabe mencionar, que en la última década se ha ido desarrollando la tendencia de utilizar seguidores de rodillo en este tipo de motores, buscando la reducción del ruido y la disminución del combustible. Un ejemplo claro lo tenemos en la tecnología desarrollada por una gran firma automovilística en el mercado, Audi, donde en su sistema Audi valvelift se aprovechan las cualidades de los mecanismos de leva para regular el consumo de combustible en función de la demanda de potencia de cada situación de la conducción [19].

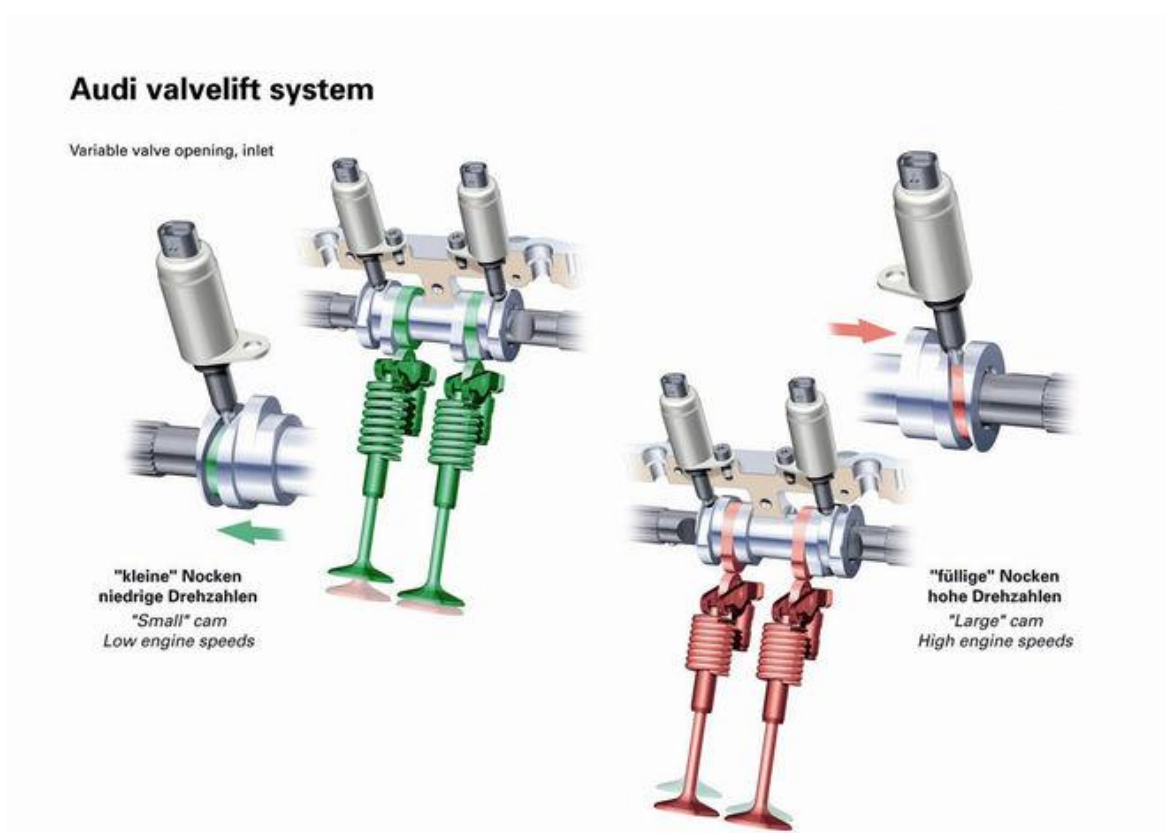


Figura 39. Sistema Audi Valvelift con uso de seguidores de rodillo.



¿Leva radial o axial?

Ante esta pregunta la elección vendrá determinada por el espacio y la geometría de la máquina donde instalaremos nuestro sistema. Si las especificaciones de diseño requieren de un movimiento de salida paralelo al eje del árbol de levas, no nos quedará otra opción que utilizar seguidores empujados por leva radial.

Si no se especifica nada en concreto, y siempre que geométricamente podamos encontrar una solución aceptable, la elección de levas radiales es favorable. Su fabricación es menos costosa y sus posibilidades de diseño son mayores, en general, es más flexible y barata.

¿Lubricar o no lubricar?

Esta pregunta tiene rápida respuesta. Las levas necesitan mucha lubricación. Aunque puede resultar una obviedad, es recomendable remarcar que este tipo de mecanismos, mientras no se especifique lo contrario, deben recibir una buena dosis de lubricante. De hecho, las levas automotrices que tanto hemos comentado en esta sección funcionan literalmente inmersas en aceite.

El lubricante, por lo general un aceite limpio con distintos aditivos, debe ser capaz de reducir la fricción en el movimiento del mecanismo y disipar todo el calor posible. Sólo en casos en los que los sistemas leva-seguidor operan cerca de un producto susceptible a ser contaminado (medicamentos, alimentos, textil...) trabajarán secas, sin lubricación.

Para concluir esta sección, y con ello este capítulo del trabajo, se añaden de forma resumida algunas de las reglas que nos brinda R. L. Norton en su libro referenciado [2]. Deben servir como directrices que el profesor nos ofrece desde su extensa y exitosa experiencia en el diseño de estos mecanismos, pero no como unas bases de obligatorio cumplimiento.

1. Trate de evitar los detenimientos innecesarios a la hora de describir la función de movimiento de la leva. Cada parada resultará ser una penalización en la dinámica de su mecanismo.
2. Cuando tenga que considerar el efecto de las vibraciones en su sistema y esté indeciso en su elección de función, escoja un movimiento polinómico 3-4-5 o una función de seno modificada antes que una función trapezoidal modificada.
3. Para reducir al máximo las vibraciones de su mecanismo donde la velocidad del árbol de levas es prácticamente constante, una función polinómica es la más adecuada.



4. El ángulo de presión debería estar limitado a 30° para seguidores de traslación convencionales, aunque se pueden tolerar valores cercanos a 35° cuando el seguidor es deslizante por bola y optimizado para baja fricción. Para seguidores oscilantes, un ángulo de presión generalmente aceptable es de 35° .
5. Recuerde que cuando utiliza seguidor de rodillo en su sistema de leva-seguidor, debe mantener un valor máximo absoluto de radio de curvatura mayor a R_r .
6. Por otro lado, cuando utilice un seguidor de cara plana, no debe permitir radios de curvatura negativos en la superficie del contorno de la leva.
7. Debe prestar especial atención en el diseño de las articulaciones entre los seguidores y su respectivo árbol de levas para tratar de minimizar o eliminar los efectos de momentos torsores y flectores que puedan aparecer durante el funcionamiento del sistema, en vistas a evitar fenómenos de fatiga en los componentes.





3. HERRAMIENTAS DE SOFTWARE PARA LA PROGRAMACIÓN





3.1. INTRODUCCIÓN

A día de hoy, existen distintas plataformas de desarrollo y entornos de programación para el sistema operativo Android, el más determinante del mercado en estos momentos. La elección del sistema operativo en el que nos apoyaremos ha sido sencilla como se explica a continuación.

Por otro lado, se ha debido realizar un estudio y un balance sobre ventajas e inconvenientes de distintas herramientas de software para la programación. La decisión tomada es coherente con el grado de conocimientos que se posee de la materia en programación, y del contexto y objetivos que se buscan con la app “Cálculo de levadas”.

3.2. ANDROID

Android es un sistema operativo basado en el núcleo Linux. Desarrollado por Android Inc., comprada por Google en 2005, se ha convertido en los últimos años en el sistema operativo número uno para teléfonos inteligentes. Su expansión es inmensa, llegando a estar presente en más de mil millones de dispositivos donde incluimos tablets, televisores, relojes e incluso coches.

Para la carretera

Android Auto se ha diseñado pensando en la seguridad. Con una interfaz sencilla e intuitiva, controles integrados en el volante y nuevas y potentes acciones de voz, permite reducir las distracciones para que puedas centrarte en la carretera.



Figura 40. Sistema Android implantado en ordenador de a bordo de un coche.

Como otras grandes firmas de la tecnología, Android surgió en el núcleo tecnológico más importante del mundo, en Palo Alto, California, donde dio sus primeros pasos con la primera de sus versiones: Android 1.0 Apple Pie, el 5 de noviembre de 2007. No sería hasta 2008 cuando el sistema operativo estaría disponible para los terminales.



Desde la empresa destacan la flexibilidad que el sistema operativo posee y la facilidad de uso para el consumidor. La plataforma es adaptable a pantallas de mayor resolución y cuenta con importantes tecnologías de conectividad útiles en el día a día como Bluetooth o Wi-Fi. Otras características importantes son el gran soporte multimedia que posee, la gran gama de soporte para hardware como cámara de fotos y vídeo, GPS, acelerómetros, sensores de proximidad y de presión, sensores de luz o la cualidad tan importante de multitarea, es decir, que las aplicaciones que no se están ejecutando en primer plano reciben ciclos de reloj, o no se interrumpen.

En Android siempre se ha cuidado la sencillez en la ejecución y a su vez lo visual, llamativo y original. Buena cuenta de ello está en sus diferentes versiones, que en inglés, reciben el nombre de diferentes postres o dulces. En cada versión, el postre o dulce elegido empieza por una letra distinta, conforme un orden alfabético:

- A. Apple Pie (v1.0), Tarta de manzana.
- B. Banana Bread (v1.1), Pan de plátano.
- C. Cupcake (v1.5), Magdalena.
- D. Donut (v1.6), Rosquilla o Dónut.
- E. Éclair (v2.0/v2.1), pastel francés conocido como pepito o canuto.
- F. Froyo (v2.2), abreviatura de “Frozen Yogurt”, Yogur Helado.
- G. Gingerbread (v2.3), Pan de jengibre.
- H. Honeycomb (v3.0/v3.1/v3.2), Panal de miel.
- I. Ice Cream Sandwich (v4.0), Sandwich de helado.
- J. Jelly Bean (v4.1/v4.2/v4.3), Gominola o judía de gominola.
- K. KitKat (v4.4), Kit Kat.
- L. Lollipop (v5.0/v5.1), Piruleta.
- M. Marshmallow (v6.0), Malvavisco o nube.



Figura 41. Versión Android 6.0 Marshmallow.



Android está considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Se ha convertido en poco tiempo en un modelo a seguir por desarrolladores de tendencia y competencia en el mercado.

Y es que Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a una lista de incidencias donde se pueden ver problemas todavía no resueltos y reportar problemas nuevos. El usuario es capaz de cambiar incluso la aplicación que usa para enviar mensajes de texto, puede añadir widgets en la pantalla de inicio o implementar launchers (estilos o temas en las pantallas de inicio).

Así no es de extrañar que Android supere cada año en la cuota de mercado a sus competidores más directos. Cabe destacar la presencia aplastante de este sistema operativo en países tan importantes como Alemania, Francia, España, E.E.U.U. o China, pero aún le queda mucho por recorrer en países orientales como Japón [15].

Smartphone OS Sales Share (%)

Germany	3 m/e Dec 14	3 m/e Dec 15	% pt. Change	USA	3 m/e Dec 14	3 m/e Dec 15	% pt. Change
Android	70.9	72.6	1.7	Android	47.6	59.1	11.5
iOS	20.9	20.2	-0.7	iOS	47.7	39.1	-8.6
Windows	7.4	6.4	-1.0	Windows	3.8	1.6	-2.2
Other	0.8	0.8	0.0	Other	0.9	0.2	-0.7
GB	3 m/e Dec 14	3 m/e Dec 15	% pt. Change	China	3 m/e Dec 14	3 m/e Dec 15	% pt. Change
Android	50.1	51.9	1.8	Android	77.0	71.4	-5.6
iOS	41.7	38.6	-3.1	iOS	21.5	27.1	5.6
Windows	7.4	9.2	1.8	Windows	0.7	1.2	0.5
Other	0.8	0.3	-0.5	Other	0.8	0.3	-0.5
France	3 m/e Dec 14	3 m/e Dec 15	% pt. Change	Australia	3 m/e Dec 14	3 m/e Dec 15	% pt. Change
Android	64.8	69.9	5.1	Android	43.7	53.5	9.8
iOS	22.9	20.5	-2.4	iOS	45.1	39.6	-5.5
Windows	11.6	8.7	-2.9	Windows	9.2	6.3	-2.9
Other	0.7	0.9	0.2	Other	2.0	0.6	-1.4
Italy	3 m/e Dec 14	3 m/e Dec 15	% pt. Change	Japan	3 m/e Dec 14	3 m/e Dec 15	% pt. Change
Android	67.3	76.9	9.6	Android	38.3	44.4	6.1
iOS	18.3	14.5	-3.8	iOS	60.2	54.1	-6.1
Windows	12.7	8.1	-4.6	Windows	0.0	0.0	0.0
Other	1.7	0.5	-1.2	Other	1.5	1.5	0.0
Spain	3 m/e Dec 14	3 m/e Dec 15	% pt. Change	EU5	3 m/e Dec 14	3 m/e Dec 15	% pt. Change
Android	83	86.3	3.3	Android	65.6	71.0	5.4
iOS	12.9	12.2	-0.7	iOS	24.5	21.2	-3.3
Windows	3.8	1.5	-2.3	Windows	9.0	7.2	-1.8
Other	0.3	0.0	-0.3	Other	0.9	0.6	-0.3

Figura 42. Ventas smartphones 2015.



Ya tenemos unas pautas claras de que la elección de Android como sistema operativo que soporte la aplicación de desarrollo de este Trabajo Fin de Grado es sin duda la mejor opción, pero aún queda comentar lo más importante. Y es que Android cuenta con una gran comunidad de desarrolladores que empujan al gran sistema hacia el avance, desarrollando hasta el día de hoy más de 1.000.000 de aplicaciones (más de dos tercios son gratuitas) disponibles en la tienda oficial: Google Play.

Sin duda alguna, Android es el sistema operativo más globalizado y demandado por desarrolladores de aplicaciones. Como hemos comentado, su código abierto facilita el trabajo de aquellos que buscan aportar un plus a este gran sistema, que a su vez comparten sus avances e ideas, como si de una gran familia se tratara.

3.3. ECLIPSE Y OTROS ENTORNOS DE PROGRAMACIÓN

Eclipse es un software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar aplicaciones y proyectos. Desde el primer momento se barajó la opción de realizar la aplicación en este entorno de desarrollo principalmente de lenguaje Java.

Se trata de una potente y completa plataforma de programación, desarrollo y compilación de elementos muy variados. Es una herramienta bien extendida y conocida por los desarrolladores, de la cual existe gran cantidad de documentación especializada.

Es un programa inicialmente desarrollada por IBM desde 2001, y actualmente gestionada por la Fundación Eclipse. El software se basa en una plataforma de cliente enriquecido, la cual está constituida por una plataforma principal, y unas características secundarias. Las funciones de Eclipse son más bien de carácter general, aunque se pueden ampliar y mejorar mediante el uso de módulos (plug-ins).

Esto es una gran ventaja, pues la concepción de módulos permite a Eclipse ser una plataforma ligera para componentes de software y además le posibilita la extensión a otros lenguajes de programación como son C/C++ o Python. A lo largo de sus diferentes versiones se ha ido mejorando la interfaz del programa y creando ayudas para los usuarios como atajos o simplificaciones en el código.



Figura 43. Software Eclipse.



Por otro lado, existen otras herramientas de programación para Android más específicas como son Android Studio o Basic4Android.

El primero es un entorno de desarrollo integrado (IDE), anunciado en 2013 en la conferencia Google I/O. Resultó relevante cuando en dicha conferencia se consideró a esta herramienta como el entorno oficial para el desarrollo de aplicaciones para Android, por delante de Eclipse. Dentro de sus características más destacables encontramos la renderización en tiempo real, plantillas para crear diseños y componentes comunes de Android y una consola de desarrollador muy útil para visualizar en todo momento el desarrollo del proyecto que se esté llevando a cabo.

Por otro lado, Basic4Android es otra plataforma alternativa de las muchas que existen para el desarrollo de apps en este sistema operativo. No es de las más utilizadas ni la visualización de su interfaz es de las más agradables, pero resulta sencilla con unos conocimientos mínimos de programación en lenguaje Java.



Figura 44. Software Android Studio.



Figura 45. Software Basic4Android.

3.4. APP INVENTOR

Una vez analizado el sistema operativo sobre el cual queremos trabajar, y habiendo visto una gama de posibilidades de entornos de programación para Android, llega la hora de seleccionar una herramienta adecuada a nuestros requisitos: buscamos un software con el que podamos desarrollar una aplicación avanzada, de una carga numérica importante, a su vez con interfaz amigable y sencilla para trabajar, y para desarrolladores con conocimientos básicos de programación pero no específicamente en lenguaje Java.

Desde luego que la respuesta es rápida: AppInventor.

Google AppInventor es una plataforma desarrollada desde Google Labs para crear aplicaciones de software para el sistema operativo Android. De forma visual, y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando y uniendo una serie de bloques para implementar la aplicación. Se trata de un sistema completamente gratuito y fácil de configurar desde su página web oficial.



La primera versión de la herramienta se publicó en julio del año 2010 y está dirigida principalmente a personas que no están familiarizadas con este campo de la informática. En sus primeros comienzos el software tuvo gran éxito y aceptación pero pronto se quedó atrás respecto a otros entornos de desarrollo. No tardaría en volver al mercado cuando la compañía decidió unir esfuerzos con el prestigioso Instituto de Tecnología de Massachusetts (MIT).

Desde entonces, el destino de esta herramienta ha cambiado de forma significativa, cuando se decidió abrir el código y se lanzó una segunda versión más completa y sin tantos fallos.



Figura 46. Software AppInventor.

Actualmente, desde la página oficial podemos encontrar gran cantidad de información y ayuda para dar los primeros pasos en la creación de aplicaciones Android. Desde el sitio web podemos acceder a interesantes tutoriales, historias, blogs, documentación, libros, fórums...que ayudarán al usuario a empezar de la mejor manera. A día de hoy, MIT AppInventor cuenta con una comunidad cercana a los 3 millones de usuarios de más de 195 países de todo el mundo.

Desde la página oficial comparten el asombroso dato de que más de 100.000 usuarios activos construyen semanalmente alrededor de 7 millones de aplicaciones Android [20]. Además, nos muestran las metas que se fijan desde la compañía como son la promulgación de la educación en ciencia computacional y la construcción de enlaces y contactos entre empresas colaboradoras.

En cuanto a su estructura se divide en dos partes bien diferenciadas pero intercomunicadas. Son el módulo “Designer” y el módulo “Blocks”.

En el primero el usuario diseña el aspecto de la pantalla desde cero, arrastrando y soltando sobre la pantalla los componentes que desea. Estos componentes a su vez se dividen en familias como son componentes de interfaz de usuario (botones, cajas de texto, sliders, check box...), componentes de organización, multimedia, sensores (acelerómetro, giróscopo,



lector de códigos QR...) y varias más. En este módulo también se puede modificar las características de los componentes, desde su color, ancho, largo, hasta su forma.

Por otro lado, cada uno de los componentes tiene asociados unos bloques de programación a modo de puzle que permitirán la creación del código. Desde el módulo Blocks, el usuario va uniendo y relacionando estas piezas para ir conformando la aplicación. A su vez existe una categoría de bloques como se muestra en la siguiente imagen.

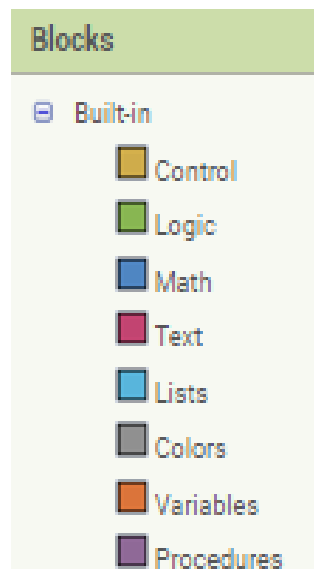


Figura 47. Categoría de bloques en AppInventor.





4. METODOLOGÍA





4.1. ETAPAS DE DISEÑO

Tal como se ha ido describiendo a lo largo de este Trabajo Fin de Grado, la finalidad del mismo es proporcionar una herramienta simple en su uso, fácil, amena a la vista e interactiva para que cualquier persona con un dispositivo Android pueda emplearla, ya sea un estudiante o un ingeniero de diseño. Dicha herramienta facilitará unos valores orientativos y aproximados del cálculo de levas planas de rotación con seguidor traslacional. De esta forma fijamos los requisitos básicos de nuestra aplicación:



Figura 48. Requisitos de diseño de la aplicación.

- ***Diseño atractivo***
Su presentación y organización invita al usuario su uso, guiándole a través de las distintas pantallas.
- ***Fácil de usar***
Se ha buscado la simpleza en cuanto a la presentación de los datos y siempre que se ha podido todos los datos del menú correspondiente se exponen en la pantalla buscando su mejor visualización.
- ***Sencillo de entender***
La aplicación se divide en tres bloques principales, donde los dos primeros ayudarán al usuario a comprender el alcance de la herramienta desde un punto de vista teórico de los sistemas leva-seguidor y otro punto de vista más funcional de la app. En el tercero de los bloques se procede al cálculo de estos sistemas de forma guiada.



- **Interactivo**

Complementado al requisito anterior, se trata de enganchar en la medida de lo posible al usuario para que pueda completar un ejercicio. Mediante ventanas emergentes y anotaciones de resultados se ayudará al usuario en todo el proceso de diseño de la leva.

4.2. ELECCIÓN DE LA HERRAMIENTA PARA EL DESARROLLO

Para la creación de la aplicación se hará uso de un programa informático conocido como ApplInventor. Ya hemos visto algunas de las alternativas posibles en cuanto a entorno de programación, y hemos dado alguna pincelada de lo que es ApplInventor.

A continuación profundizaremos en la herramienta elegida, viendo los aspectos más valiosos de la misma, los que permiten desarrollar aplicaciones de un contenido variado y amplio con facilidad e incluso intuición. También comentaremos algunos de los inconvenientes de ApplInventor, pues al ser un software relativamente joven tiene muchas cosas que pulir.

Introducción a ApplInventor

En secciones anteriores se ha tratado de dar una visión más histórica del software utilizado, destacando la importancia que tuvo la introducción del Instituto de Tecnología de Massachusetts, sirviendo como lanzadera para este entorno.

Ahora se busca detallar el cómo empezar a utilizar ApplInventor desde cero, aportando una visión de los diferentes recursos que se han encontrado para ello. Además, una vez dentro de la plataforma, veremos bien cómo se estructura el sistema y cómo manejar los diferentes bloques para ir creando las aplicaciones.

La primera, y más importante fuente donde encontrar información básica y ayuda para desenvolverse por ApplInventor es su sitio web oficial [20]. Desde la página se divulga una extensa cantidad de información acerca del mundo ApplInventor, desde una amplia descripción de todos los bloques y funciones que forman las aplicaciones hasta foros donde se plantean problemas de programación y uso y se tratan de dar solución.

Cabe destacar la importante promulgación que se le da al conocimiento de desarrollo de aplicaciones desde la web, animando a los usuarios a compartir sus creaciones, premiando a los más originales tanto en diseño como en funcionalidad de sus trabajos. También podemos encontrar un listado de eventos y conferencias programadas que incentivan a la gente a comenzar en este apasionante mundo. Desde luego, ApplInventor es más que un simple software, es una gran familia.

Una vez inmersos en esta web, es hora de comenzar a programar. En la página nos enseñan a configurar nuestro teléfono y nuestro ordenador para poder testear los



avances que vayamos realizando en tiempo real. Se nos ofrecen diversas opciones, tanto si tenemos un dispositivo con sistema operativo Android como si no:



Figura 49. Conexión al teléfono vía Wi-Fi.



Figura 50. Uso de emulador en el ordenador.

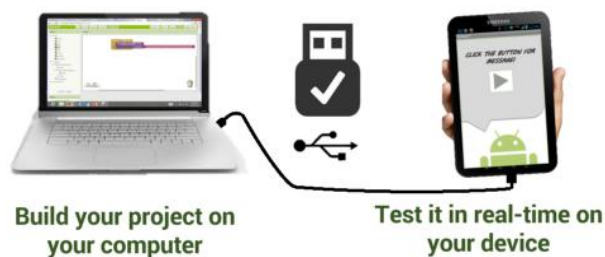


Figura 51. Conexión al teléfono vía USB.



Como vemos, no es estrictamente necesario disponer de un dispositivo Android, pues ya sea mediante un emulador o vía USB, podemos probar el funcionamiento de nuestras apps en tiempo real. Desde la web nos recomiendan utilizar la primera opción en el caso de tener un dispositivo que cuente con sistema Android.

Tras elegir la vía de desarrollo que más nos convenga podemos comenzar a programar. Para ello sólo necesitaremos una cuenta de Google (Gmail). La versión que utilizaremos será la versión 2 de AppInventor, que periódicamente se va mejorando y corrigiendo.

Una vez dentro, se ha de iniciar un nuevo proyecto que nos llevará a la pantalla de diseño de aplicaciones: “Designer”, desde donde iremos arrastrando los componentes que queramos a la mesa de trabajo (representada como una pantalla de un teléfono móvil o tablet), y podremos configurar las propiedades de los mismos. También se irán organizando todos los componentes que vayamos utilizando de forma esquematizada.

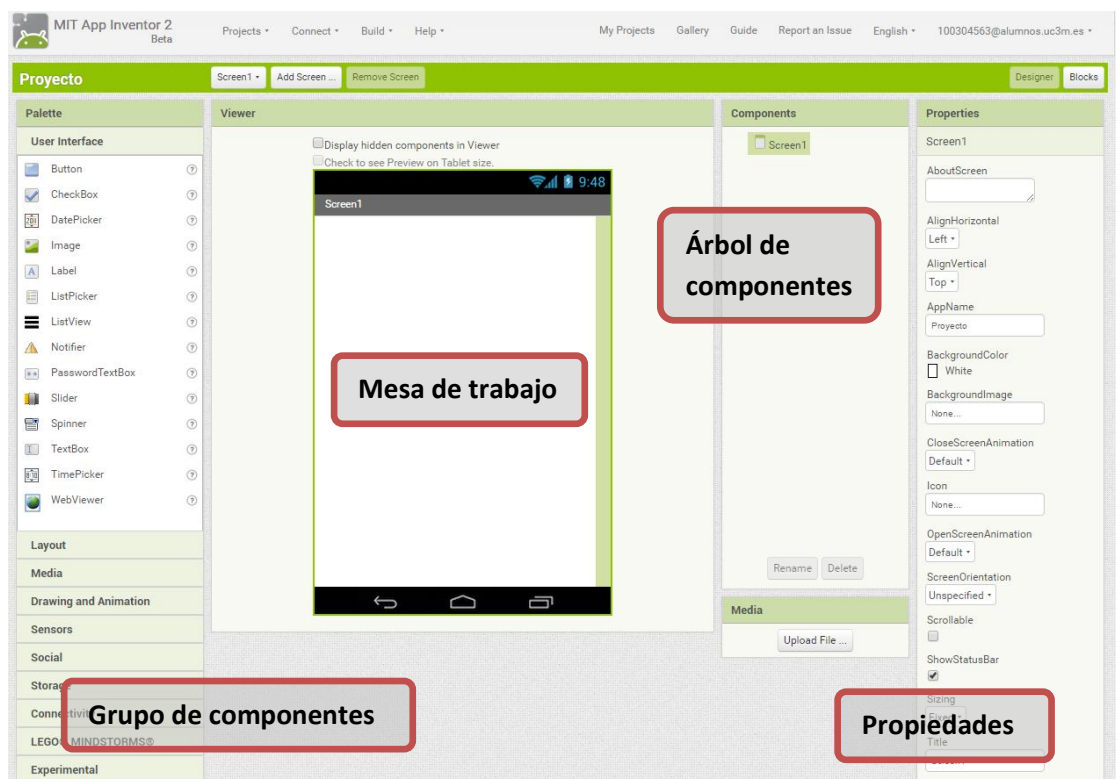


Figura 52. Pantalla “Designer”.

Es en esta primera pantalla donde se comenzará dando forma a nuestra aplicación. Podremos trabajar la estética a través de las propiedades de los componentes y organizar su localización dentro de la pantalla del móvil de forma fácil y rápida.



Cuando tengamos un primer montaje de los componentes que formarán parte de la aplicación pasaremos a darle lógica a la misma a través de la pantalla “Blocks”. Se trata de ir uniendo bloques, que serían las líneas de código en otro software convencional, para ir conformando la aplicación y dando relaciones entre los componentes.

En dicha pantalla aparecerá de nuevo el árbol de componentes mostrado en la pantalla anterior además de unos bloques temáticos comunes para cualquier componente. Son los vistos en la Figura 47. Clickando sobre cualquier componente del árbol de componentes se desplegará una serie de bloques asociados a funciones del mismo. Por ejemplo, si introducimos un botón en nuestro proyecto, un bloque que nos aparecería en su ventana desplegable sería aquel que da comienzo a una acción cuando se pulse sobre el botón.



Figura 53. Pantalla “Blocks”.

De esta forma, vemos como a pesar de ser dos pantallas independientes, existe la unión entre el módulo “Designer” y el módulo “Blocks” a través del árbol de componentes. También podemos observar en las dos figuras anteriores que se trata de una forma de desarrollo de aplicaciones muy amigable y fácil de entender, con una interfaz de lo más intuitiva. A priori todo son ventajas, pero veremos cómo conforme va creciendo la aplicación surgen inconvenientes con los que debemos competir.



Ventajas e inconvenientes de ApplInventor

Sin duda alguna se puede destacar ApplInventor por la facilidad que ofrece al desarrollador para la creación de aplicaciones Android. Sin necesidad de conocer un lenguaje de programación específico, como puede ser Java, cualquier usuario de ApplInventor puede ser capaz de confeccionar una app.

Como hemos visto, la página web oficial sirve de fuente de recursos para comenzar y aprender el manejo del software, pero cuando el desarrollador quiere realizar aplicaciones más amplias y complejas es cierto que la web se queda corta. Desde otros espacios en internet se pueden encontrar algunos cursos online que te introducen en ApplInventor, pero no hasta un nivel muy avanzado.

Por lo tanto, ApplInventor es ideal cuando la necesidad es una aplicación sencilla, que apenas se tarde en programar unos minutos y sea fácil de usar por cualquier consumidor. Aun así, el grado de avance de la aplicación dependerá del desarrollador, y ha quedado demostrado que se pueden realizar aplicaciones extensas y a la vez funcionales.

Tal como se ha comentado antes, la forma de programar las aplicaciones es mediante la unión de unos bloques a modo de puzle. La gran ventaja de esto es la posibilidad de aplicar la intuición a la hora de programar, además de que gracias al código de colores que utiliza el software, la creación de aplicaciones se hace rápida y sencilla.

Pero el uso de los bloques trae consigo algunas desventajas que pueden ser motivo de descarte de la herramienta de trabajo, y es que son limitados. El desarrollador se debe ceñir a los bloques existentes y cuantos más utilice, más lenta irá la aplicación y más difícil será el trabajo, colapsándose por momentos la interfaz. Cabe destacar, que la organización en la mesa de trabajo de la pantalla “Blocks” es fundamental, pues al no existir un orden de líneas es fácil perder bloques durante el trabajo por todo el espacio que ofrece ApplInventor.

Afortunadamente, el crecimiento de ApplInventor es muy rápido. Aparece un gran salto cualitativo entre la primera y la segunda versión del software, donde se incluyen nuevos bloques y la forma y visualización de la interfaz es más cómoda y agradable para trabajar. Cada vez hay más variedad en los bloques que se pueden utilizar y se ofrecen más facilidades a programadores inexpertos que quieren introducirse en este mundo de desarrollo de aplicaciones Android.

Otras desventajas que se pueden encontrar en ApplInventor frente a otros entornos de programación es que no se da la posibilidad de hacer aplicaciones multitarea, es decir, en cuanto a la posibilidad de ejecutar acciones simultáneamente, conocido en otros lenguajes de programación como “hilos de ejecución”. Esto ralentiza de forma exponencial el avance de la aplicación cuando se deben hacer muchos cálculos para pasar de una pantalla a otra, por ejemplo. A tenor de esto último, cabe destacar que cuando el proyecto que se está realizando supera las diez pantallas, ApplInventor lanza un mensaje de aviso y no asegura que la aplicación vaya a compilar satisfactoriamente.

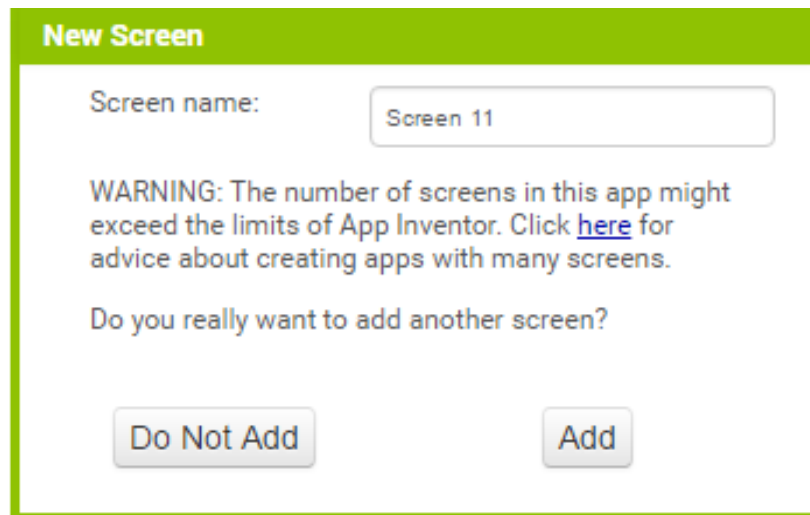


Figura 54. Aviso superación 10 pantallas.

Para concluir, una de las grandes ventajas de AppInventor a parte de la facilidad que ofrece al público en general para implementar apps es su uso desde el navegador web. Trabajar en internet y no sobre un software descargable y ejecutable permite mantener el trabajo en un lugar seguro en la web, siempre asociado a la cuenta de Gmail. Además, en cualquier momento, se puede descargar el proyecto que se desee en formato *.apk* o en extensión *.aia* (propia de AppInventor), permitiendo a su vez importar proyectos desde el ordenador al espacio de trabajo.

4.3. DESARROLLO DEL PROYECTO

Una vez fijada la temática a desarrollar (cálculo y diseño de levass), el mercado en el que queremos introducirnos (Android) y la herramienta con la que vamos a desarrollar nuestra aplicación para dicho sistema operativo (AppInventor) es hora de conocer las referencias que han inspirado la aplicación, el método seguido para el diseño y las herramientas que han hecho posible el resultado final.

Antes de programar la aplicación, primero realizaremos una aplicación de prueba para comprobar el nivel gráfico y de cálculo al que se puede llegar programando en el entorno que nos ofrece AppInventor.

Referencias

La referencia principal que se ha utilizado para el desarrollo de la aplicación “Cálculo de levas” ha sido un pequeño, pero potente software proporcionado por el departamento de Ingeniería Mecánica de la Universidad Carlos III de Madrid: SC Levas.

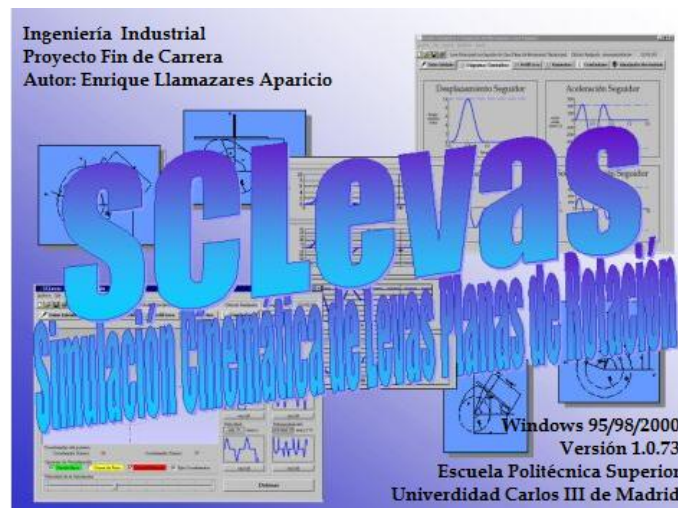


Figura 55. Software SC Levas.

El software desarrollado como Proyecto Fin de Carrera por Enrique Llamazares Aparicio, es una herramienta de interés para el cálculo, diseño y simulación cinemática de levas planas de rotación. De manera simple, el usuario puede seleccionar las características de leva plana que desea calcular y desplazarse por diferentes menús que se proporcionan tras el cálculo.

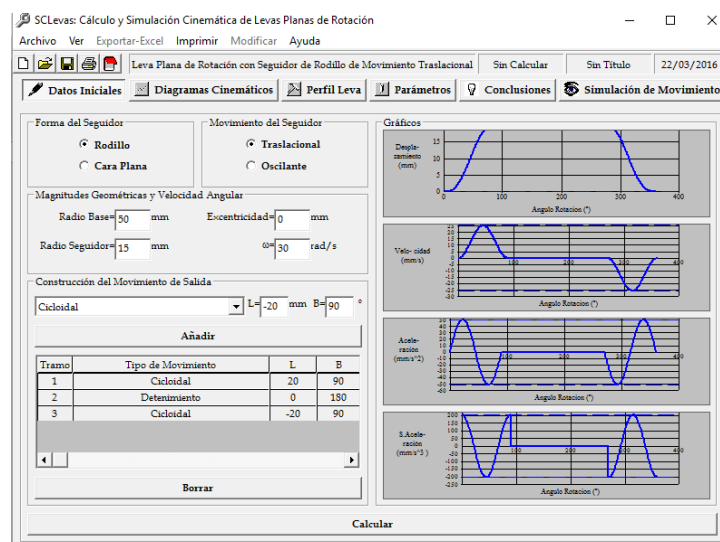


Figura 56. Pantalla principal SC Levas.



El programa cuenta con opciones tanto de visualización como de tratamiento de resultados que lo hacen especialmente atractivo para el usuario, sirviendo como gran fuente de inspiración para el presente Trabajo Fin de Grado.

De esta manera se intentará imitar la filosofía de SC Levas para la implementación de la aplicación objeto de este trabajo. Realmente la dinámica de la aplicación no se debe alejar mucho. Se trata de ofrecer al usuario la posibilidad de diseñar una leva de forma sencilla a través de unos pasos simples. Mostrar los resultados de forma clara y útil para el usuario es clave para el éxito de la aplicación.

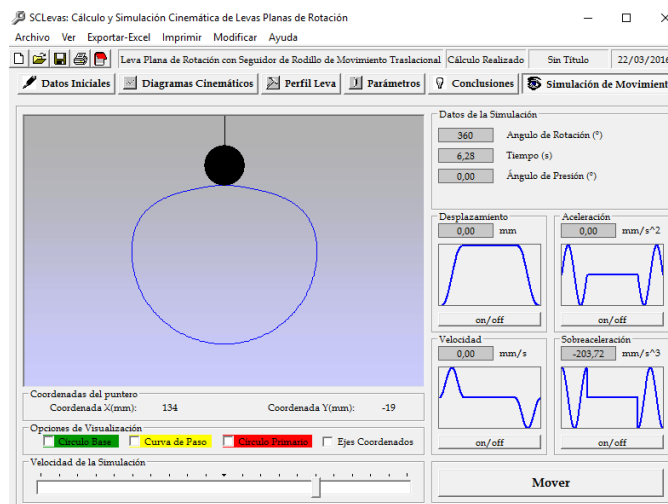


Figura 57. Pantalla visualización sistema leva-seguidor.

Además, al igual que en el software de referencia, se permitirá al usuario la exportación de los resultados a una hoja de cálculo Excel, y aprovechando las funcionalidades que nos aporta ApplInventor y la facilidad de intercomunicación de los dispositivos móviles, smartphones y tablets, se buscará la posibilidad de compartir estos archivos a través de aplicaciones externas a la app “Cálculo de levas”.

Diseño de la aplicación

Como hemos visto, SC Levas sirve de gran precedente para el diseño de la app “Cálculo de levas”. Dicha aplicación pretende ser la primera herramienta educativa multiplataforma de la materia desarrollada para un mercado tan amplio y mundial como es el de los smartphones y tablets. Desarrollado para Android, el sistema operativo por excelencia para móviles y dispositivos smart.

El primer paso para el desarrollo de la aplicación ha sido conformar su estructura. Resulta fundamental tener claro desde un primer momento los contenidos que se pretenden abordar con la aplicación. Bien es cierto que durante el desarrollo de la misma, la programación puede sufrir variaciones, pues como se ha explicado, ApplInventor no es una herramienta perfecta y presenta muchas limitaciones para aplicaciones multipantalla.



El esqueleto de la aplicación fijado desde un primer momento es:

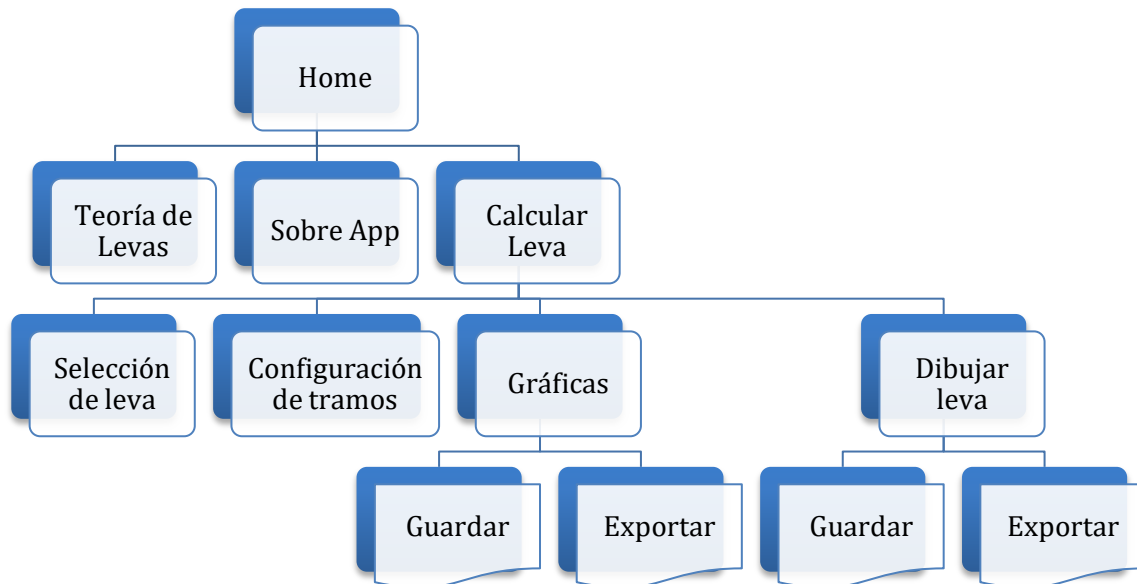


Figura 58. Organigrama de la aplicación.

La idea era realizar una secuencia de pantallas que nos llevasen a un resultado útil y que nos proporcionase además información sobre las levass que pudiésemos utilizar fuera de la aplicación. Conociendo las limitaciones de AppInventor, no se ha podido profundizar demasiado en cuestiones de programación, por lo que supone un reto desde el principio realizar este trabajo de forma sencilla y optimizando los recursos.

Una vez determinada la estructura del proyecto se procede a programar la parte gruesa de la aplicación, el menú “Calcular Leva”. En la primera pantalla de este menú, el usuario debe fijar los parámetros iniciales que conformarán la leva. En primer lugar se selecciona el número de tramos, inicialmente limitado a seis, y la forma del seguidor, que puede ser circular, puntual o plano. Posteriormente, se ajustarán mediante unos sliders las magnitudes geométricas y la velocidad angular en r.p.m. necesario para el cálculo de las gráficas.



La problemática que surgió en esta pantalla fue la elección de los componentes que utilizaríamos para seleccionar y configurar la leva. ApplInventor cuenta con un limitado grupo de componentes que llevan a cabo estas actividades, de entre los cuales no se encuentra un Radio Button que habría sido ideal para el caso de la selección de número de tramos y seguidor.

Para ello se utilizó el componente CheckBox. En la parte de Blocks fue necesario el diseño de un algoritmo con bucles “if” para que en todo momento solo se pudiese seleccionar una opción y no dos o más a la vez.

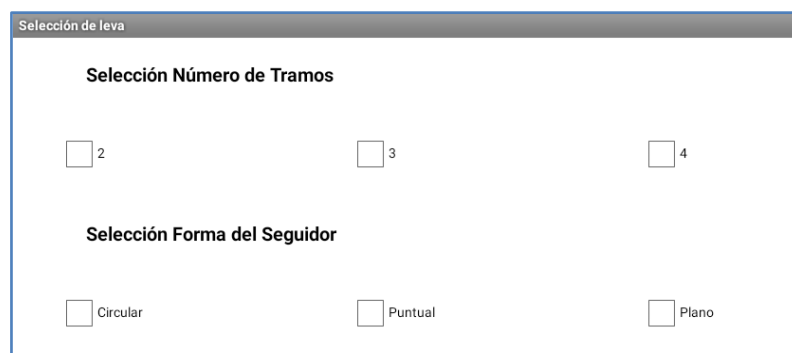


Figura 59. Uso de componentes CheckBox.

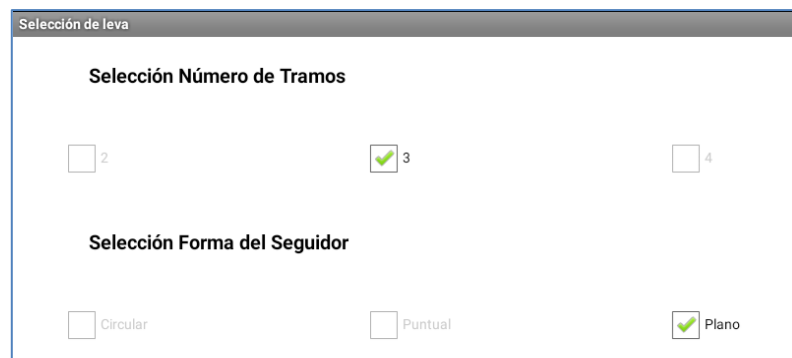


Figura 60. Selección de tramos y tipo de seguidor.

Por otro lado cabe destacar que las magnitudes geométricas que aparecerán por pantalla son en función del seguidor. En el caso de seleccionar seguidor circular será necesario establecer unos valores de radio de seguidor y de excentricidad en mm. Éste último sólo será considerado a efectos de cálculos para la determinación del ángulo de presión más adelante.

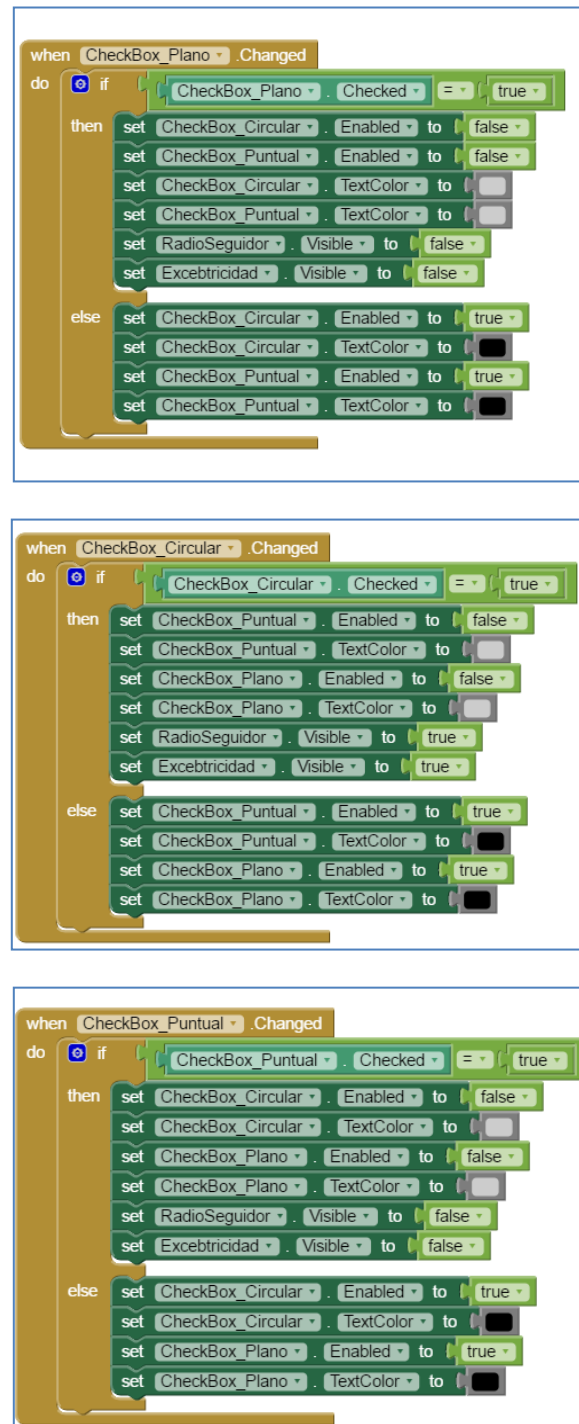


Figura 61. Algoritmos para seleccionar únicamente un CheckBox.

Establecidos todos los datos de entrada deseados es hora de pasar a la siguiente pantalla. Aquí configuraremos el número de tramos que previamente se ha decidido. Será



necesario introducir al dispositivo la altura recorrida y el ángulo de rotación de cada tramo mediante TextBox, además de elegir el tipo de curva del tramo a través de un ListPicker.

Se da la opción de elegir entre curva armónica, cicloidal, velocidad constante y detenimiento. Cabe mencionar que en un primer momento también se daba la posibilidad de curva polinómica 3-4-5 pero AppInventor era incapaz de empaquetar el proyecto al saturarse y se decidió suprimir esta opción, al igual que reducir el número de tramos de seis a cuatro.

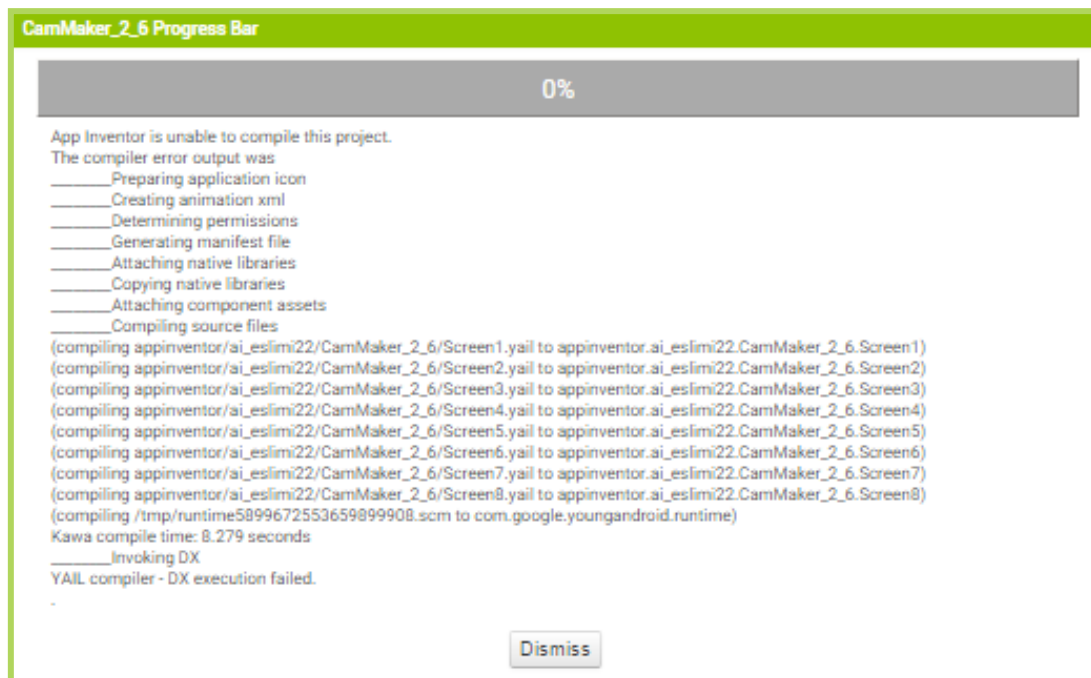


Figura 62. Error de compilación del proyecto.

Tanto en la pantalla anterior como en esta, es posible que el usuario trate de avanzar sin rellenar algún campo o violando la ley de la continuidad de la leva. Para evitar esto, se han programado una serie de avisos mediante bucles elementales que proporciona el software, explicándolos a continuación:

- En el caso de no rellenar un campo, una pantalla emergente nos indicará que no podemos continuar sin completar todos los campos.
- Si se ha seleccionado detenimiento en el tipo de curva, la altura recorrida en el tramo correspondiente debe ser nula, de no ser así seremos avisados.
- Se ha de tener en cuenta que la suma total de las alturas debe ser nula para que la leva tenga una gráfica de desplazamientos continua.
- La suma total de los ángulos recorridos en los tramos debe ser 360° , de lo contrario la leva estaría incompleta.



- En último lugar, la gráfica de desplazamientos debe estar en todo momento por encima del eje horizontal, es decir, no se puede tener una altura menor a cero. Si se diese el caso, de nuevo una ventana emergente nos avisaría para corregir el error.

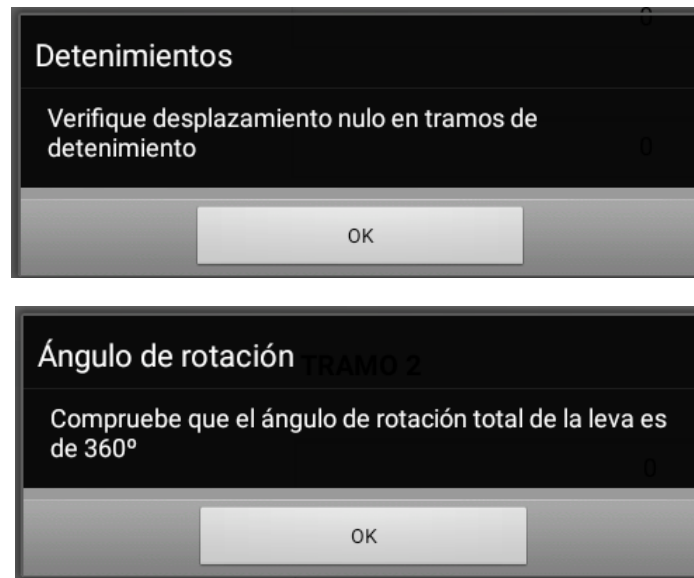


Figura 63. Ejemplos ventanas emergentes de ayuda

Una vez cerciorados de que los datos introducidos son correctos es en la siguiente pantalla donde visualizaremos las gráficas. Sin duda es esta pantalla la que más programación requiere. En ella se han desarrollado todas las ecuaciones que describen las curvas para cualquier tramo. Puesto que Applinventor no cuenta con bloques matemáticos de derivación, se han tenido que montar todas las derivadas una a una de cada una de las curvas para cada tramo.

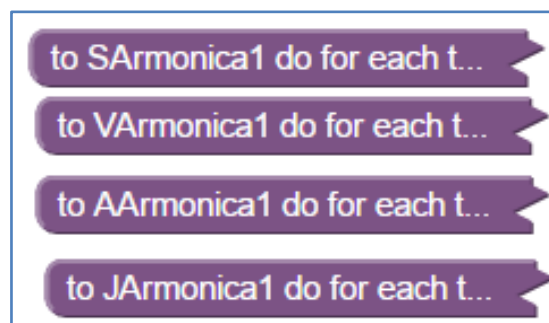


Figura 64. Funciones de curvas y derivadas usadas en la aplicación.



Para hacer esto de forma correcta y ordenada, se han desarrollado funciones (bloques morados en AppInventor) que contienen las ecuaciones correspondientes. En las funciones de las curvas podemos encontrar un bucle “for” que recorre un índice θ desde 0 hasta el valor del ángulo recorrido por el tramo, introducido previamente. Para cada valor de θ dibujamos sobre el componente Canvas un punto, hasta completar el trazado del tramo.

Estas ecuaciones contienen la función algebraica de cada tipo de curva con un sumando que la sitúa de forma correcta en el Canvas y un término de factor de escala que multiplica a la función para agrandar o disminuir su visualización. Además, guardamos los valores en vectores que utilizaremos para cálculos varios. Éstos deben ser vectores de números enteros redondeados, para que no existan incompatibilidades a la hora de detectar la continuidad de la función global en las gráficas de velocidad, aceleración y sobreaceleración. Desafortunadamente perderemos precisión para la hora de representarla en programas externos. Además, incluiremos bucles “if” en los bloques de las derivadas cinemáticas para detectar máximos y mínimos en los vectores definidos.

Se ha de destacar los pocos bloques de dibujo que presenta AppInventor, aportando únicamente bloques de dibujo de rectas introduciendo punto de inicio y fin, círculos, puntos y texto. Para conseguir un efecto de trazo continuo en la curvas, se ha aumentado el grosor de los puntos.

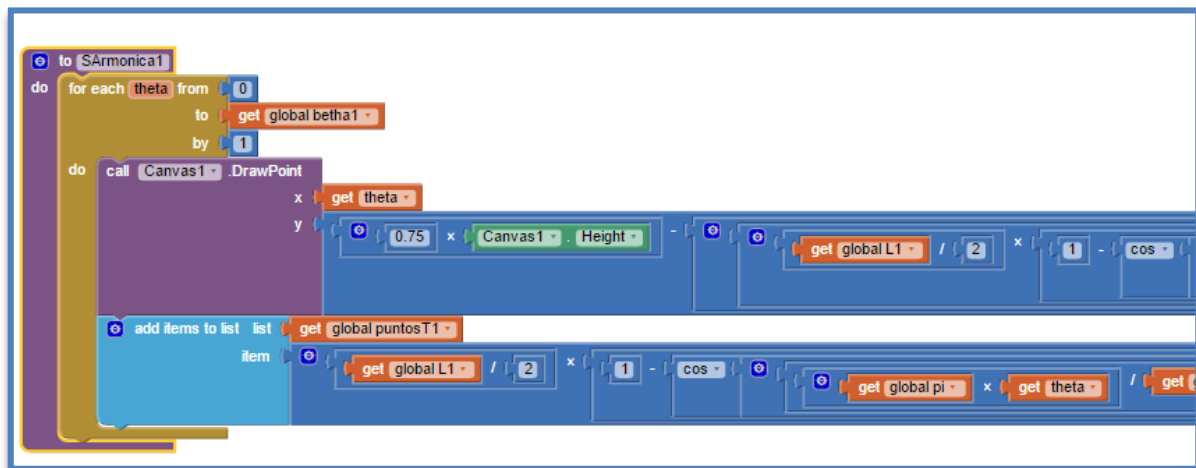


Figura 65. Detalle función armónica simple de desplazamiento.



Otras funciones importantes que se han implementado son:

- *PintarCurvas*: conociendo el número de tramos y la curva seleccionada anteriormente por el usuario, desde esta función se hace la llamada a las funciones que dibujan los tramos sobre los Canvas. De esta manera, al llamar a dichas funciones se van rellenando los vectores con los puntos deseados como se ha comentado antes.
- *LímiteMáximoYMínimo*: esta función aprovecha los vectores obtenidos de las curvas que han sido graficadas para obtener los valores máximos y mínimos de las gráficas. Estos valores se escribirán en el componente Canvas correspondiente.
- *VaciarVectores*: función que borra los vectores que almacenan los valores de desplazamiento, velocidad, aceleración y sobreaceleración de cada tramo.
- *VectoresGlobales_S_V_A_J*: en esta función se concatenan los vectores parciales de cada tramo, de tal manera que se devuelven vectores que contienen todos los puntos de las gráficas. Obtenemos cuatro vectores: PuntosS, PuntosV, PuntosA y PuntosJ que utilizaremos para otros cálculos.
- *VaciarVectoresGlobales*: función que vacía los vectores anteriores.
- *Ejes*: en esta función se contienen otras funciones auxiliares que dibujan una recta vertical y otra horizontal en cada Canvas que servirán como ejes de ordenadas y de abscisas para las gráficas. Además se escribirá sobre cada Canvas un título que lo identifica y se completarán los valores de ángulo recorrido de cada tramo en el eje de abscisas.
- *ContinuidadVelocidad*, *ContinuidadAceleracion* y *ContinuidadSobreaceleracion*: estas tres últimas funciones reciben el número de tramos que ha sido seleccionado y los vectores parciales de cada tramo que contienen los puntos de velocidad, aceleración y sobreaceleración respectivamente. Mediante bloques específicos de tratamiento de vectores de ApplInventor comparamos el último valor del vector de un tramo con el primer valor del tramo siguiente, de la misma manera lo hacemos entre el último valor del último tramo y el primer valor del primer tramo para cerrar el ciclo. Si todos estos valores coinciden se devuelve un 1, si no un 0.

Para que podamos ver todos los resultados nada más pasar a esta pantalla, se deben realizar diversas actividades al iniciar la misma. Para esto utilizamos un bloque de control (bloques marrones) específico de ApplInventor, que ejecuta una serie de acciones que nosotros programemos en el momento de iniciar la pantalla.

Dentro de esta serie de actividades tenemos que diseñar unos algoritmos que permitan autoajustar las gráficas a los Canvas, un gran atractivo de la aplicación. Para esto necesitamos invocar a *PintarCurvas* para obtener los vectores de puntos, obtener los máximos y mínimos y realizar un algoritmo que devuelva un factor de escala determinado para unos márgenes del 10% en gráfica de desplazamiento y del 20% en las demás por encima y por debajo.

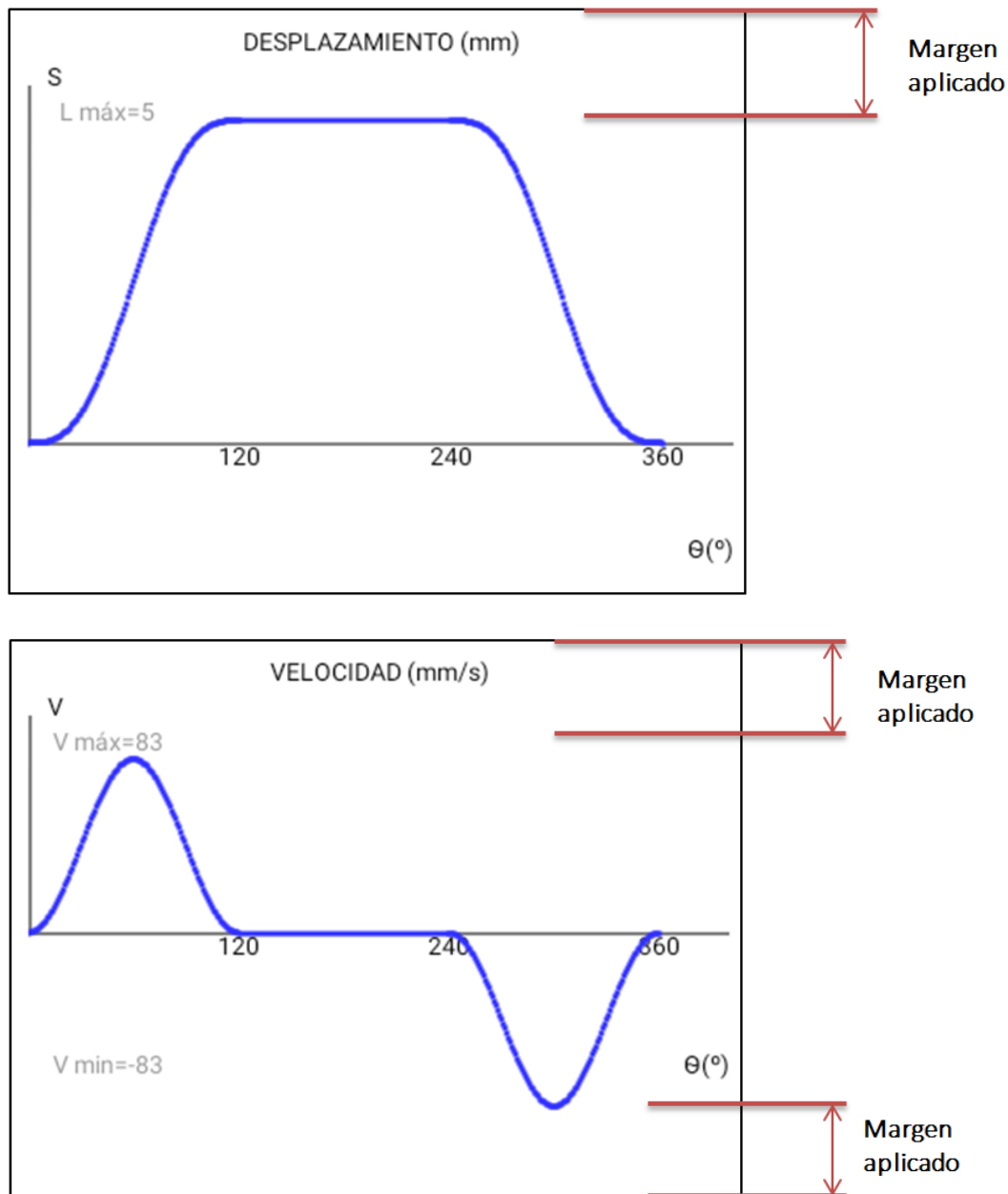


Figura 66. Márgenes en las gráficas cinemáticas.

Una vez calculado el factor de escala que permite los márgenes anteriores, es necesario llamar a la función `VaciarVectores` y de nuevo a la función `PintarCurvas`.

Para completar la vista de la pantalla de "Gráficas", se llamarán a las funciones de continuidad y mediante un algoritmo basado en bucles sencillos "if" mostrarán por pantalla en un `TextBox` si las gráficas de velocidad, aceleración y sobreaceleración son continuas o no, utilizando colores para su mejor visualización.



Una vez ejecutado el bloque inicial podemos ya visualizar la página con los diagramas representados. Como es de esperar, la transición desde la pantalla de “Configuración de tramos” a esta va a ser lenta, pues el bloque de control que se debe ejecutar antes de abrir la pantalla es largo y como hemos comentado AppInventor no da la opción de hilos de ejecución.

Para completar los modos de visualización de las gráficas, se da la opción al usuario de variar los factores de escala a su gusto. Para ello basta con pulsar sobre el botón “Ajuste manual” que habilitará unos componentes llamados Label que sirven para introducir datos al sistema. Con el botón “Ejecutar” se aplican los nuevos factores de escala por lo que será necesario borrar las gráficas de los Canvas, llamar a VaciarVectores y PintarCurvas. Si el usuario desea regresar a un modo de visualización de autoajustado presionará el botón “Autoajustar” que se habría activado al variar los factores de escala por el ajustado manual.



F.E. - S	F.E. - V	F.E. - A	F.E. - J
36.9	1.20482	0.04677	0.0009

Ajuste manual Ejecutar Autoajustar

Figura 67. Configuración escalado.

En esta pantalla, el usuario también podrá calcular las gráficas de “Ángulo de presión” (sólo para seguidor circular) y “Radio de curvatura” (para seguidores circular y plano). La forma de programación de estos dos importantes aspectos de las levas es similar. Se han realizado mediante algoritmos que calculan los valores máximos y mínimos de las ecuaciones conocidas, y los utilizan para autoajustar las gráficas a sus contenedores, de forma similar que con las gráficas anteriores. Una vez hecho esto, se muestra por pantalla el trazado de la curva como resultado de color naranja.

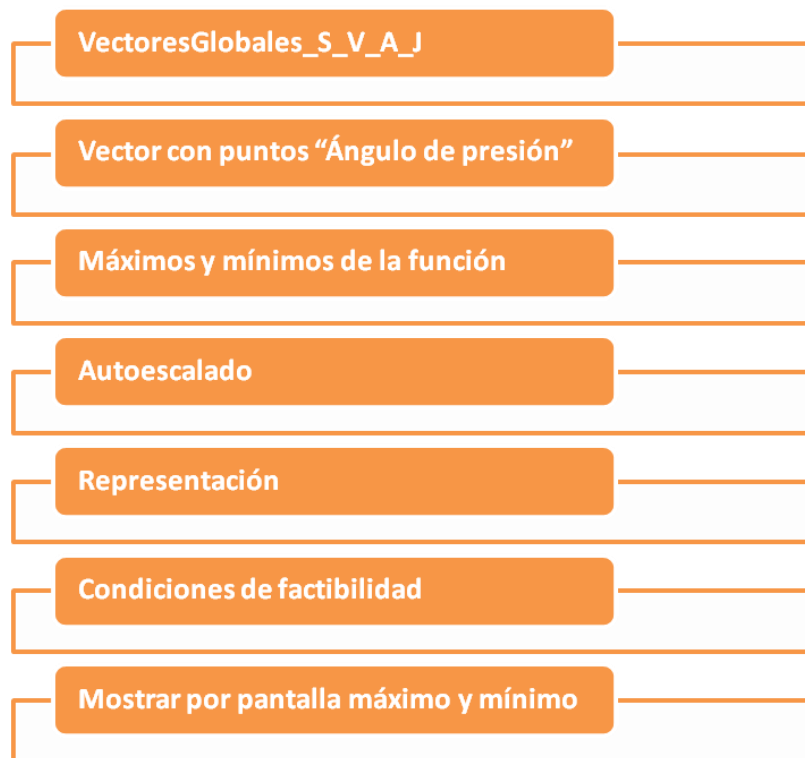


Figura 68. Pasos implementación "Ángulo de presión".

Por último, antes de pasar a la siguiente pantalla, se nos ofrece la opción de guardar y exportar. El botón "Guardar" utiliza una serie de bloques implementados en AppInventor que permiten guardar los Canvas como imagen en un directorio del dispositivo. Un mensaje emergente nos avisará en que carpeta han sido guardadas las imágenes y en que extensión.

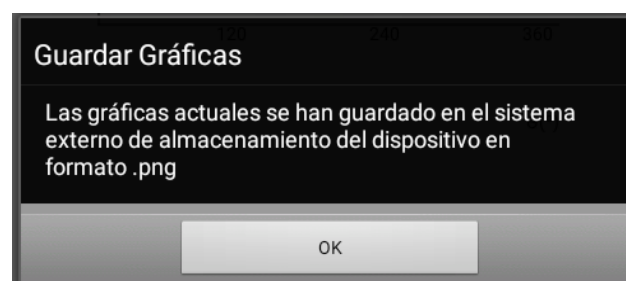


Figura 69. Ventana emergente al pulsar sobre "Guardar".

Pulsando sobre el botón "Exportar" se activará un evento o acción que mediante bloques predeterminados del entorno y el desarrollo de algoritmos y bucles, guardando los datos de las gráficas S, V, A y J en un documento. Para ello se utiliza el componente File proporcionado por el software.



Un punto muy positivo de AppInventor es la facilidad que le presenta al programador o al desarrollador de la aplicación para realizar sencillas operaciones con bloques simples y sin apenas carga de programación. Como hemos visto, cuenta con una serie de componentes que permiten sacar el máximo partido a las aplicaciones. Es el caso del componente Sharing, que permite compartir archivos generados en la aplicación utilizando aplicaciones externas instaladas en el dispositivo. De esta manera el archivo de datos se pueden enviar por correo electrónico y manipular los resultados con software externos.

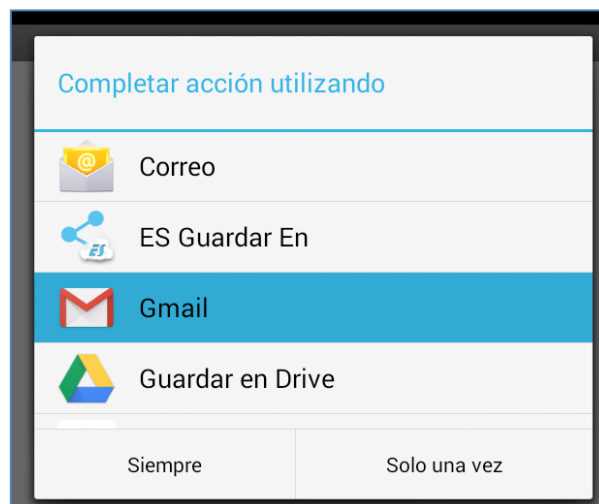


Figura 70. Compartir documentos.

Cabe destacar algunos problemas y limitaciones de AppInventor llegados a este punto, pues para aplicaciones grandes multipantalla donde existen pantallas que utilizan muchos recursos la interfaz se ralentiza de forma considerada, dificultando el trabajo al desarrollador. Además obliga al trabajador a adaptarse al entorno y a sus componentes y bloques, como el caso del componente Canvas, el cual tiene su origen de coordenadas en la esquina superior izquierda, con el eje “y” positivo hacia abajo. Esto ha condicionado mucho las expresiones de las fórmulas que se han de representar, pues todas tienen que tener un término que las ajuste de forma adecuada al contenedor como hemos visto antes.

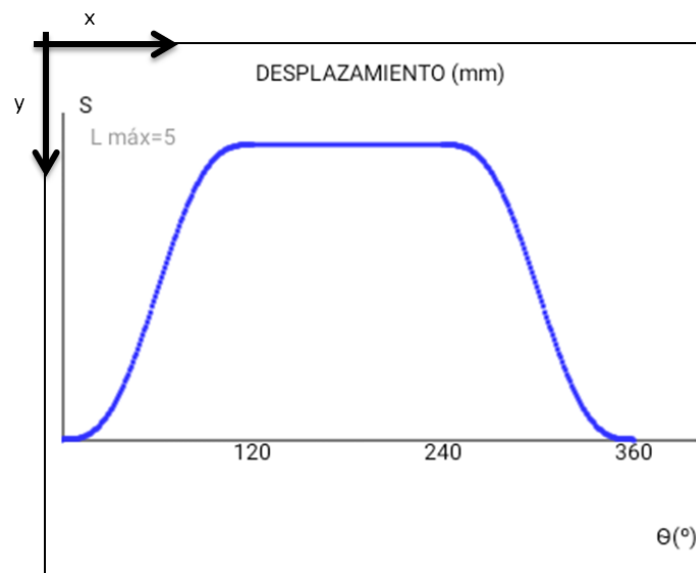


Figura 71. Origen de coordenadas en Canvas.

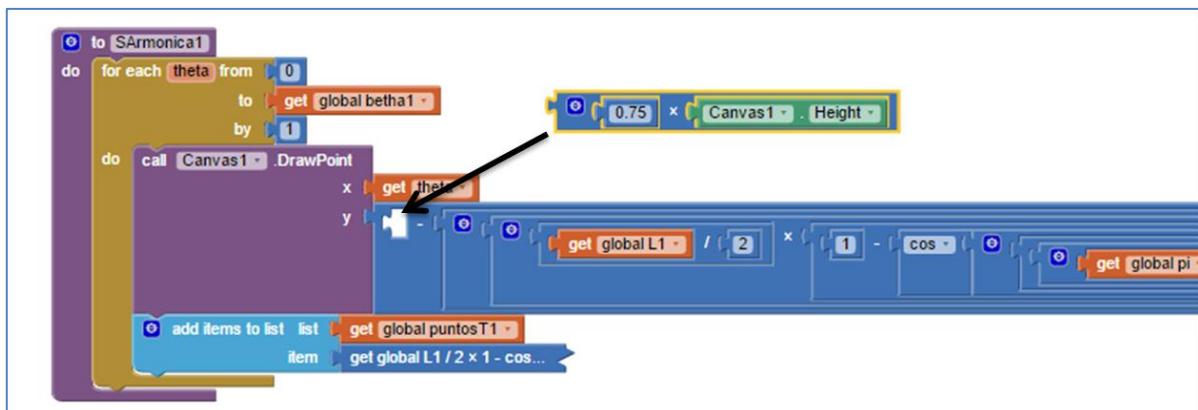


Figura 72. Bloque para situar función correctamente sobre el componente Canvas.

Además, otra limitación que presenta el componente Canvas es la imposibilidad de hacer scroll sobre él, es decir, arrastrar la pantalla hacia arriba o hacia abajo pulsando sobre él. La solución encontrada ha sido dejar márgenes entre los bordes de la pantalla y el componente, junto con el diseño de unas imágenes que indiquen esta función en los márgenes.

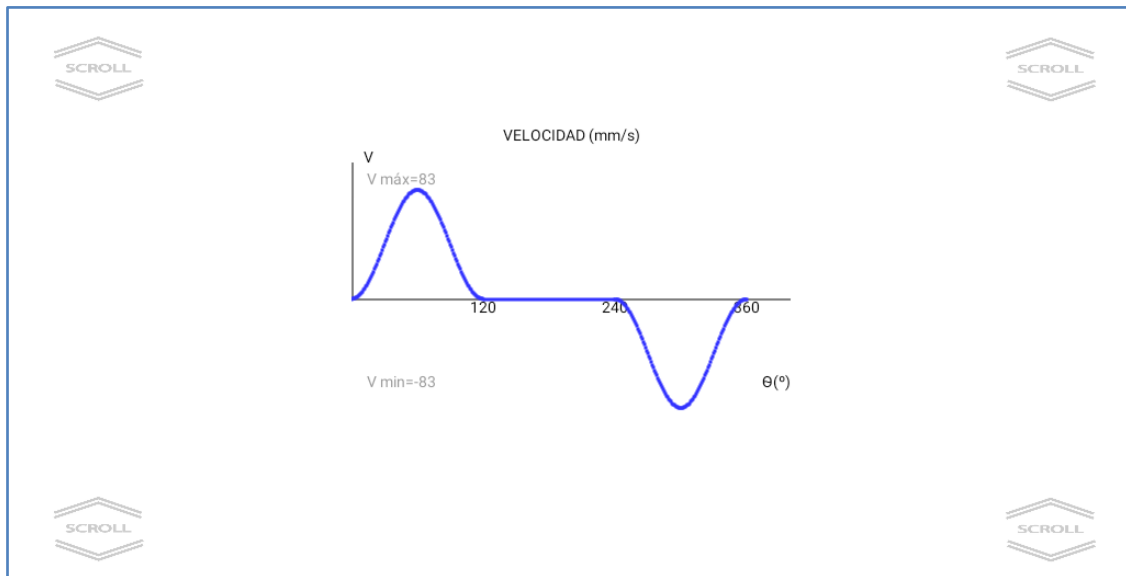


Figura 73. Indicación de scroll para pantalla “Gráficas”.

Para continuar con la estructura de nuestra aplicación, programaremos la pantalla de “Representación de leva”. En esta pantalla se pretende mostrar el perfil de la leva que se ha ido construyendo a lo largo de la aplicación. Para ello utilizaremos los vectores parciales de cada tramo que contienen los puntos del diagrama de desplazamientos.

De nuevo, sobre un componente Canvas nos disponemos a pintar dos rectas ortogonales entre sí que servirán como ejes de coordenadas. En esta pantalla se han implementado una serie de funciones que serán invocadas cuando se pulse sobre el botón “Dibujar Leva”. Se ha desarrollado una función para dibujar cada tramo. Éstas, mediante algoritmos de programación, primero distinguen si el tramo se trata de un detenimiento o no. Una vez identificado el movimiento, con el uso de bucles “while” e “if”, se va representando punto por punto del vector utilizando variables internas que van aumentando en una unidad. Éstas son el ángulo recorrido y la posición del vector de puntos del tramo.

Para conseguir cierta calidad en esta pantalla, en los algoritmos de representación de puntos se ha incluido un factor de escala que permite una correcta visualización de la leva por muy grande o pequeña que ésta pueda ser.

Por último cabe destacar la implementación de otras funciones que representan la forma del seguidor, también autoescalado. En este caso se presentó el reto de implementar una función que permitiese obtener el perfil de leva de un seguidor plano. Este perfil surge como la envolvente de la cara plana del seguidor en cada posición. Para ello, se hizo un estudio previo de cómo se podía llevar a cabo esto, escribiendo sobre el papel las ecuaciones de la recta que representaría la cara plana del seguidor en función del ángulo de rotación de la leva.

Analizaremos las expresiones calculadas a través de la siguiente figura:

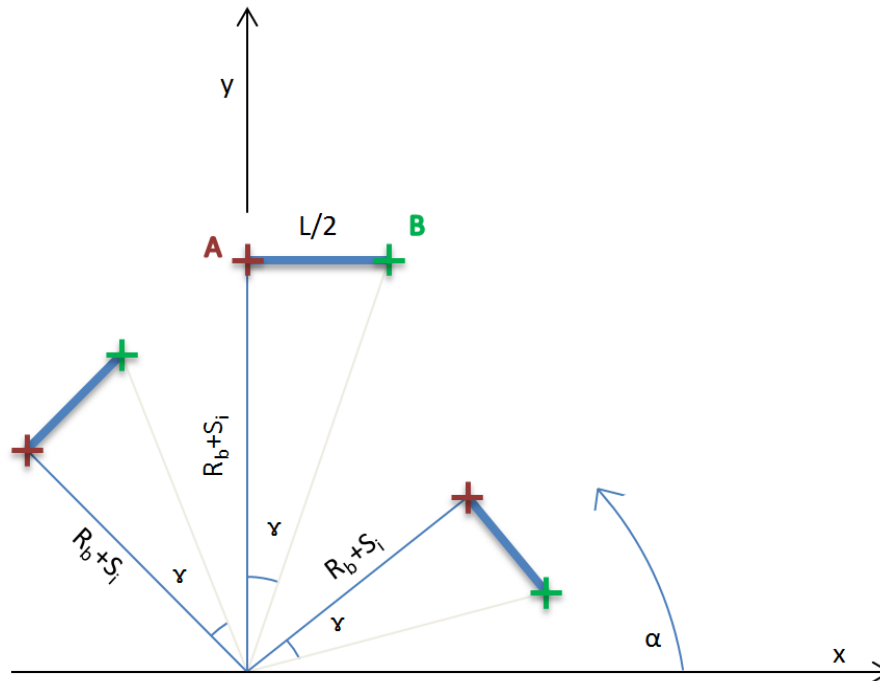


Figura 74. Representación gráfica de la implementación de las ecuaciones de seguidor de cara plana.

Tomaremos como estudio para extraer las ecuaciones de un seguidor de cara plana la semilongitud de la cara del seguidor, representada en la Figura X como la distancia entre A y B. Debemos sacar las ecuaciones en ejes cartesianos, para poder realizar la representación en App Inventor, de los puntos A (punto central de la cara del seguidor) y de B (extremo derecho de la cara). Observando el dibujo tenemos:

$$x_A = (R_b + S_i) \cdot \cos(\alpha)$$

$$y_A = -(R_b + S_i) \cdot \text{sen}(\alpha)$$

Siendo:

R_b : Radio base de la leva

S_i : Elevación del seguidor para cada i , donde $i \in [0; 360]$

α : Ángulo de rotación de la leva

Por otro, las coordenadas del punto B serán:

$$x_B = \sqrt{(L/2)^2 + (R_b + S_i)^2} \cdot \cos(\alpha - \gamma)$$

$$y_B = -\sqrt{(L/2)^2 + (R_b + S_i)^2} \cdot \text{sen}(\alpha - \gamma)$$

Donde tendríamos por trigonometría que:

$$\gamma = \arctan\left(\frac{L/2}{R_b + S_i}\right)$$

De esta forma podríamos proceder a representar el seguidor de cara plana en ApplInventor. Para dibujar la semilongitud izquierda que faltaría sabemos que es análogo pero orientado en el otro sentido, las ecuaciones de A serían iguales que las de B pero sumando α a γ en lugar de restando. Así tendríamos la leva definida de forma aproximada por las tangentes interiores que forman la totalidad de las caras del seguidor en todas las posiciones de α .

Una vez calculadas las expresiones de estas rectas en función de valores conocidos, mediante un bucle “for” se hace la representación de las caras. Para que sobre el dibujo se pueda apreciar mejor, se han pintado 60 caras planas de longitud 40 mm en lugar de 360 caras.

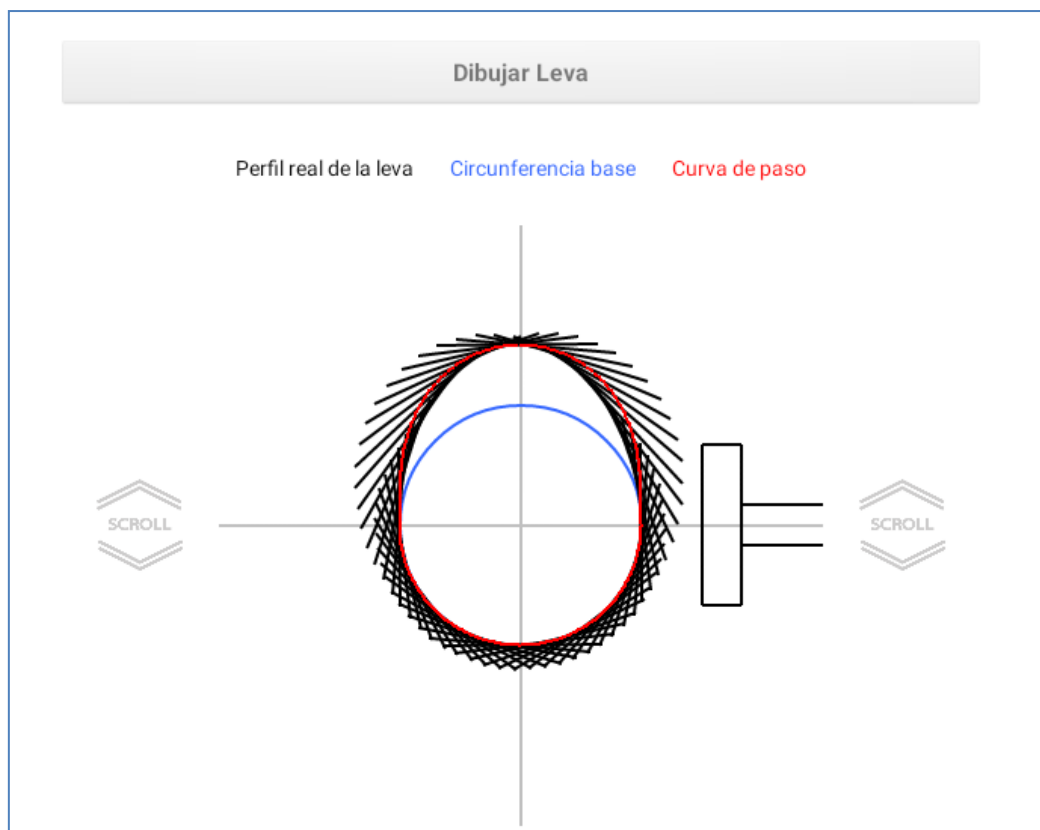


Figura 75. Representación leva plana rotacional con seguidor de cara plana.



En esta última pantalla también nos encontraremos las funciones de “Guardar” y “Exportar”, como gran atractivo de la aplicación. De forma similar a la pantalla anterior, cuando pulsamos sobre “Guardar”, salvaremos el dibujo de la leva en formato *.png* en el dispositivo. Por otro lado, cuando pulsemos sobre “Exportar”, se guardará en el directorio de descargas de nuestro dispositivo un archivo con extensión *.csv* que contiene las coordenadas cartesianas de los puntos del contorno de la leva. Como antes, también se da la opción al usuario de compartir el archivo generado.

Para que la aplicación funcione de manera correcta, cabe destacar el uso de un componente presente en la gran cantidad de pantallas, y esencial para este tipo de aplicaciones multipantalla. Se trata del componente TinyDB, una base de datos volátil al cerrar la aplicación.

Para poder transmitir información, tanto cualitativa como cuantitativa a través de las pantallas, la aplicación cuenta con cinco pequeñas bases de datos. Son componentes no visibles, no aparecen en la pantalla final, que almacenan datos dentro de la aplicación. Tienen asociadas una serie de bloques que facilitan su uso y el intercambio de datos entre pantallas, donde cada una de las pantallas que desee almacenar datos o extraerlos de estos almacenes debe contar con uno de estos elementos con el mismo nombre.

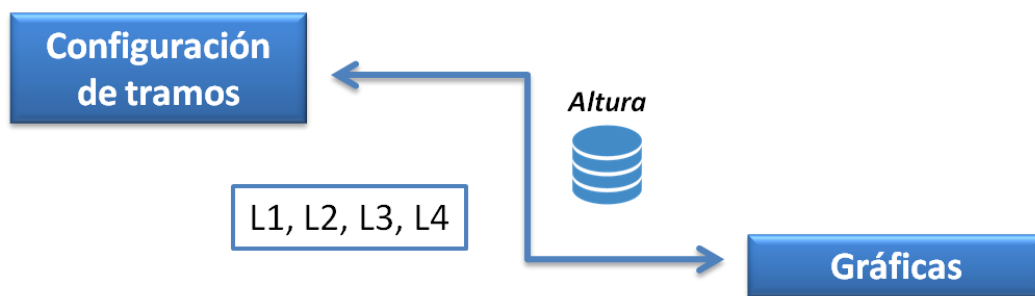


Figura 76. Representación base de datos en Applinventor.

Las bases de datos que utilizadas en la aplicación son:

- *NumeroDeTramos*: recoge el número seleccionado por el usuario en la pantalla de “Selección de Leva”.
- *Datos*: almacena distintos valores necesarios para el desarrollo de la aplicación como los valores numéricos que se ajustan en los slider de radio base, velocidad angular, excentricidad...También guarda la forma del seguidor que ha sido seleccionado.
- *Altura*: almacena la altura recorrida en cada tramo.
- *Angulo*: almacena el valor del ángulo parcial de cada intervalo.



- *Curva*: en él se contienen los valores de curva seleccionados por el usuario para cada tramo.
- *Puntos*: almacena los vectores de los puntos de desplazamiento de cada tramo que han sido completados en la pantalla “Gráficas”. Se utilizarán para dibujar la leva en la última pantalla del menú “Calcular”.

Una vez programado el gran menú “Calcular”, y con soltura ya en el entorno de programación, la implementación de los menús “Teoría de levas” y “Sobre app” no presenta mayor dificultad. Ambos menús tienen una estructura parecida basada en Layouts, componentes de AppInventor para organizar y estructurar el contenido de las pantallas. En estos podemos incluir imágenes y texto y modificar los parámetros de estos elementos para conseguir una visualización adecuada dentro de nuestra aplicación, buscando no sólo una buena funcionalidad, sino también un atractivo estético.

Requisitos estéticos

La app “Cálculo de levas” no sólo pretende ser una herramienta práctica para el diseño y cálculo de levas planas de rotación con seguidor traslacional como se ha comentado, si no que busca serlo de una forma atractiva a los ojos del usuario. Para ello se ha cuidado de forma minuciosa el diseño de los botones y el logo que completan la aplicación, siempre dentro de las limitaciones que presentaba AppInventor.

La carta de presentación de toda aplicación es su logo o icono. Inspirándonos en los estándares marcados por los desarrollos de aplicaciones Google, lo que ellos denominan Material Design, se ha tratado de constituir un logo a la altura.



Figura 77. Material Design - Icons.

Desde la web principal de Google Design podemos acceder a su sección de Material Design, donde encontraremos pautas para poder programar y diseñar cualquier tipo de acción y componentes en un estilo “Google”. Para nuestro caso buscamos diseñar un logo sencillo, atractivo y explicativo de lo que nos vamos a encontrar en la aplicación.



El software utilizado para la construcción de estos elementos será Adobe Illustrator 2015, de los desarrolladores Adobe Systems, es un editor de gráficos vectoriales que trabaja sobre un tablero de dibujo o comúnmente llamado “mesa de trabajo” y está destinado a la creación artística de dibujo y pintura para ilustración. Es una herramienta relativamente sencilla de aprender con multitud de opciones que nos permitirá aportar un toque estético importante a nuestra aplicación.

Decidida la herramienta, Adobe Illustrator, y teniendo las pautas que nos ofrece Material Design, buscamos la inspiración en algunos de los iconos más conocidos de Google.



Figura 78. Ejemplo iconos Google Design.

Algunas de las instrucciones que se nos aporta para el diseño de los iconos son las formas y dimensiones del contorno del logo, las capas que debe contener, bordes y sombreado, color...es así como tras un proceso de pruebas e intentos llegamos hasta el logo final de la app “Cálculo de levass”.



Figura 79. Pruebas previas icono.

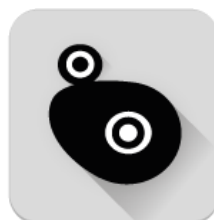


Figura 80. Icono final Material Design.

Dentro de Material Design también encontramos una extensa explicación de cómo diseñar lo que ellos denominan System Icons, un pequeño icono que simboliza un comando o acción como por ejemplo borrar, pintar o guardar. Su diseño es simple, moderno, amigable y en



ocasiones peculiar. Cada icono se reduce a su mínima forma, siempre guardando la esencia de lo que transmite. Deben ser claros e identificables incluso a un tamaño reducido.

Es así como surgen los iconos que podemos encontrar a lo largo de la aplicación, utilizados como botones de movimiento entre pantallas o como inicio de eventos ya sea guardar o exportar.



Figura 81. System Icon – Teoría de levas.

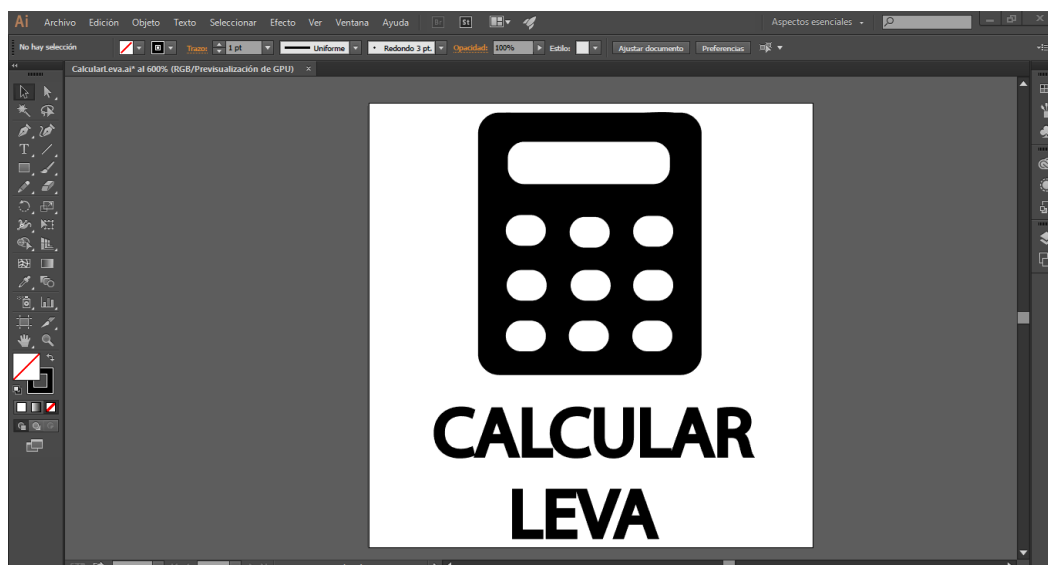


Figura 82. Diseño System Icon en Adobe Illustrator.

Para hacer aún más llamativa y visual la aplicación, nada más abrirla por primera vez accedemos a una pantalla donde nos encontramos dos pequeñas animaciones. Para realizarlas se han superpuesto una serie de 13 imágenes que crean el efecto de movimiento ya que ApplInventor no admite archivos de extensión .gif. La secuencia de imágenes ha sido desarrollada también en Adobe Illustrator 2015.

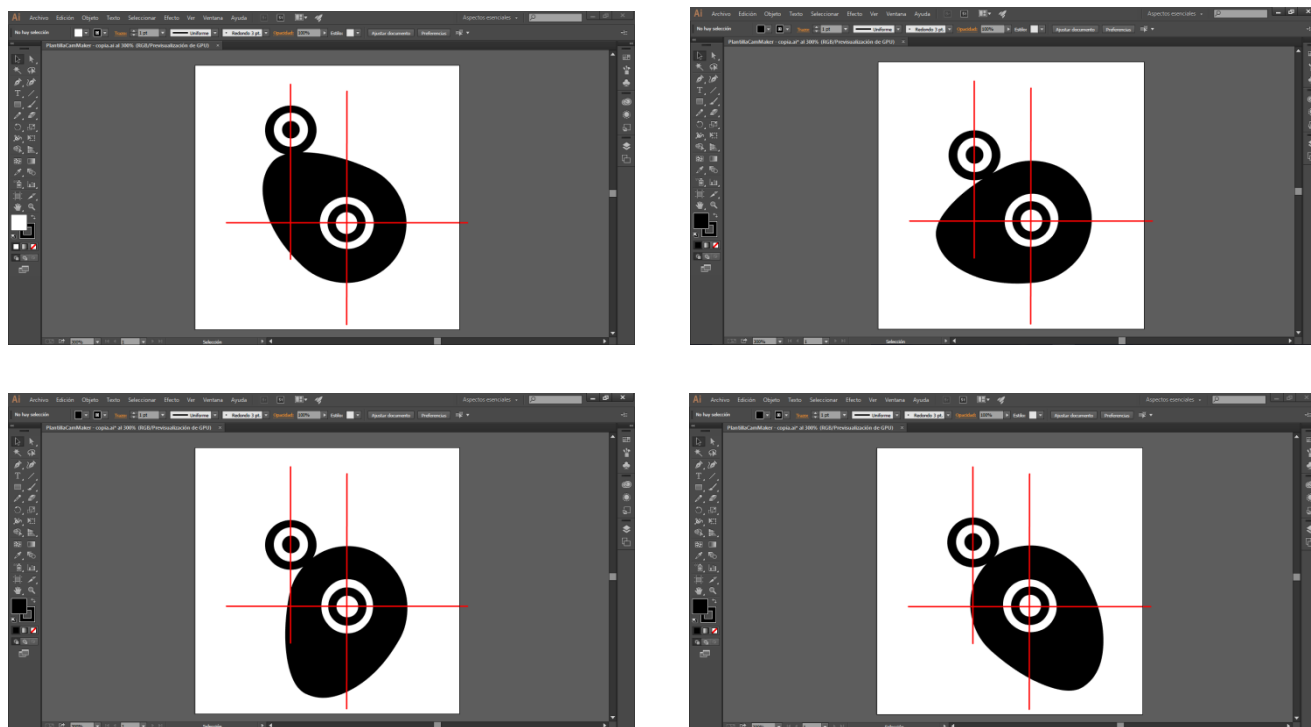


Figura 83. Diseño animación de leva en Adobe Illustrator.





5. ANÁLISIS DE RESULTADOS. PRUEBA DE FUNCIONAMIENTO DE LA APLICACIÓN





5.1. PRUEBA DE FUNCIONAMIENTO

Con el fin de comprobar el correcto funcionamiento de la aplicación, en el presente capítulo se trata de resolver un ejercicio completo de diseño y cálculo de un sistema de leva rotacional con seguidor traslacional de rodillo y ver así el alcance que posee la app “Cálculo de levas”.

De la misma forma, la resolución del ejercicio permite al desarrollador comprobar que no existen errores sustanciales en la programación de la aplicación y le ayuda a solucionarlos y a proponer posibles mejoras que serán plasmadas en trabajos futuros.

El siguiente ejercicio será resuelto en una tablet Galaxy Tab A con versión 5.0.2 de Android. Este dispositivo ha servido durante todo el proceso de desarrollo de la aplicación, por lo que la visualización está optimizada para el mismo. De todas formas, con la opción de AppInventor de crear aplicaciones con “responsive size”, la aplicación es perfectamente multiplataforma adaptando el contenido de las pantallas a las dimensiones de cada dispositivo.

Ejercicio de prueba:

Representar los diagramas de desplazamiento, velocidad, aceleración y sobreaceleración de mecanismo de leva plana rotacional con seguidor de rodillo cuya expresión analítica de la función desplazamiento en alejamientos y acercamientos es:

$$y(\theta) = L \cdot \left[\frac{\theta}{\beta} - \frac{1}{2\pi} \cdot \text{sen} \left(2\pi \frac{\theta}{\beta} \right) \right]$$

Que sigue un programa de movimiento S-D-B-D (alejamiento, detenimiento, acercamiento, detenimiento) de $\beta_1=80^\circ$, $\beta_2=120^\circ$, $\beta_3=80^\circ$ y $\beta_4=80^\circ$. Comprobar la factibilidad del mecanismo, y de no ser factible proponer una solución.

Datos: $L=5$ mm.; $R_b=25$ mm.; $R_f=7$ mm.; $w=30$ r.p.m.; $\varepsilon=0$ mm.



Para resolver el ejercicio, iniciaremos la aplicación “Cálculo de levass” y entraremos en el menú de cálculo “Calcular leva”.

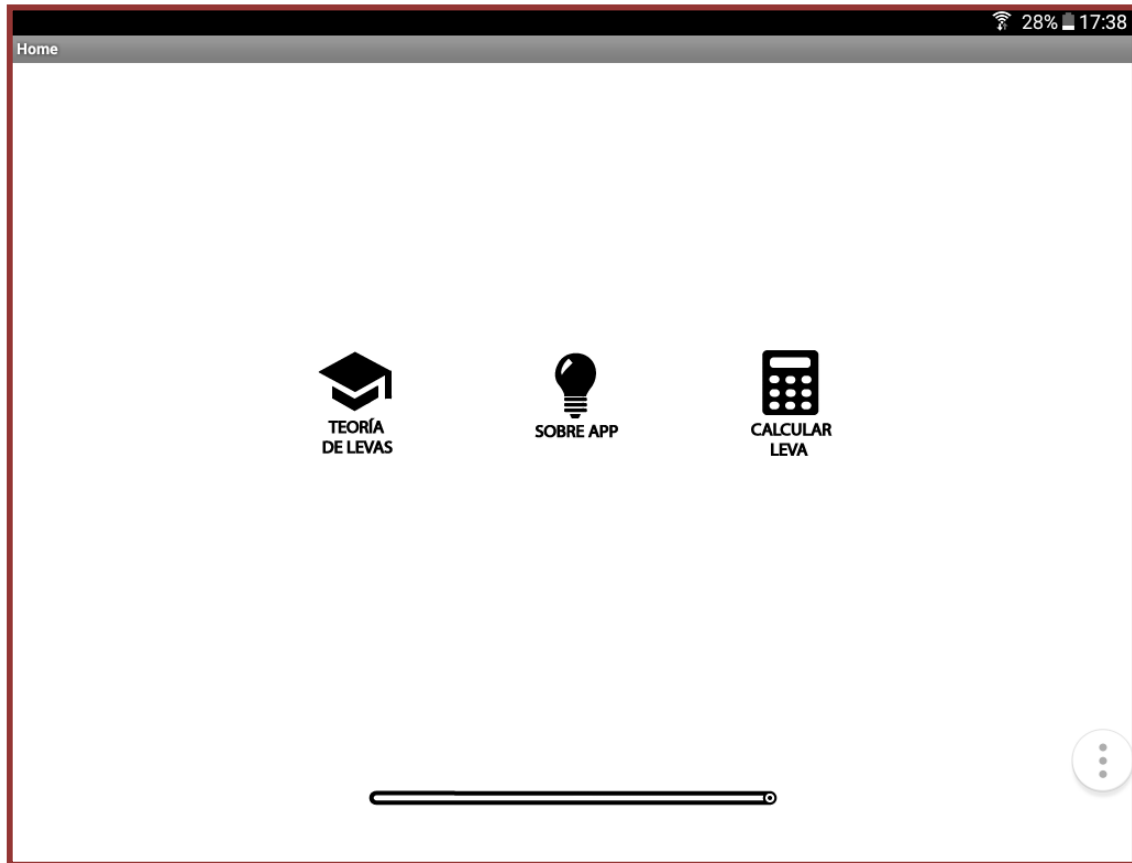


Figura 84. Pantalla “Home”.

Lo primero que debemos establecer es el número de tramos que tendrá nuestra leva, el tipo de seguidor que se va a utilizar, y en función de éste los datos que nos propone el enunciado.



Selección de leva

Selección Número de Tramos

☐ 2 ☐ 3 ☒ 4

Selección Forma del Seguidor

☒ Circular ☐ Puntual ☐ Plano

Magnitudes Geométricas y Velocidad Angular

Radio Base/Primitivo (mm) 25

Velocidad Angular (rpm) 30

Radio Seguidor (mm) 7

Excentricidad (mm) (a efectos de cálculo) 0

ATRÁS SIGUIENTE

Figura 85. Pantalla “Selección de leva”.

En la siguiente pantalla configuraremos los tramos. Cada tramo tiene como parámetros de entrada la altura recorrida, [mm], el ángulo recorrida, [°], y el tipo de curva. El enunciado del problema nos proporciona la ecuación analítica seguida en los tramos de acercamiento y alejamiento. En este caso se trata de un movimiento cicloidal, el cual está presente en la app “Cálculo de levas”.



Configuración de tramos 28% 17:30

TRAMO 1

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva

TRAMO 2

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva

TRAMO 3

Altura recorrida (mm)

Ángulo recorrido (°)



Tipo de curva

TRAMO 4

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva

ATRÁS GRÁFICAS




Figura 86. Pantalla “Configuración de tramos”.



Cabe destacar que para los detenimientos la altura recorrida del tramo debe ser nula y que para los alejamientos introducimos un valor negativo que permita cumplir la ley de continuidad de la leva, por ello $L_3 = -5$ mm.

Una vez completados todos los campos avanzamos a la pantalla “Gráficas”, donde visualizaremos las representaciones cinemáticas del movimiento, cambiando los factores de escala para mejorar la visualización si fuese necesario, y podremos observar si la leva es factible o no.

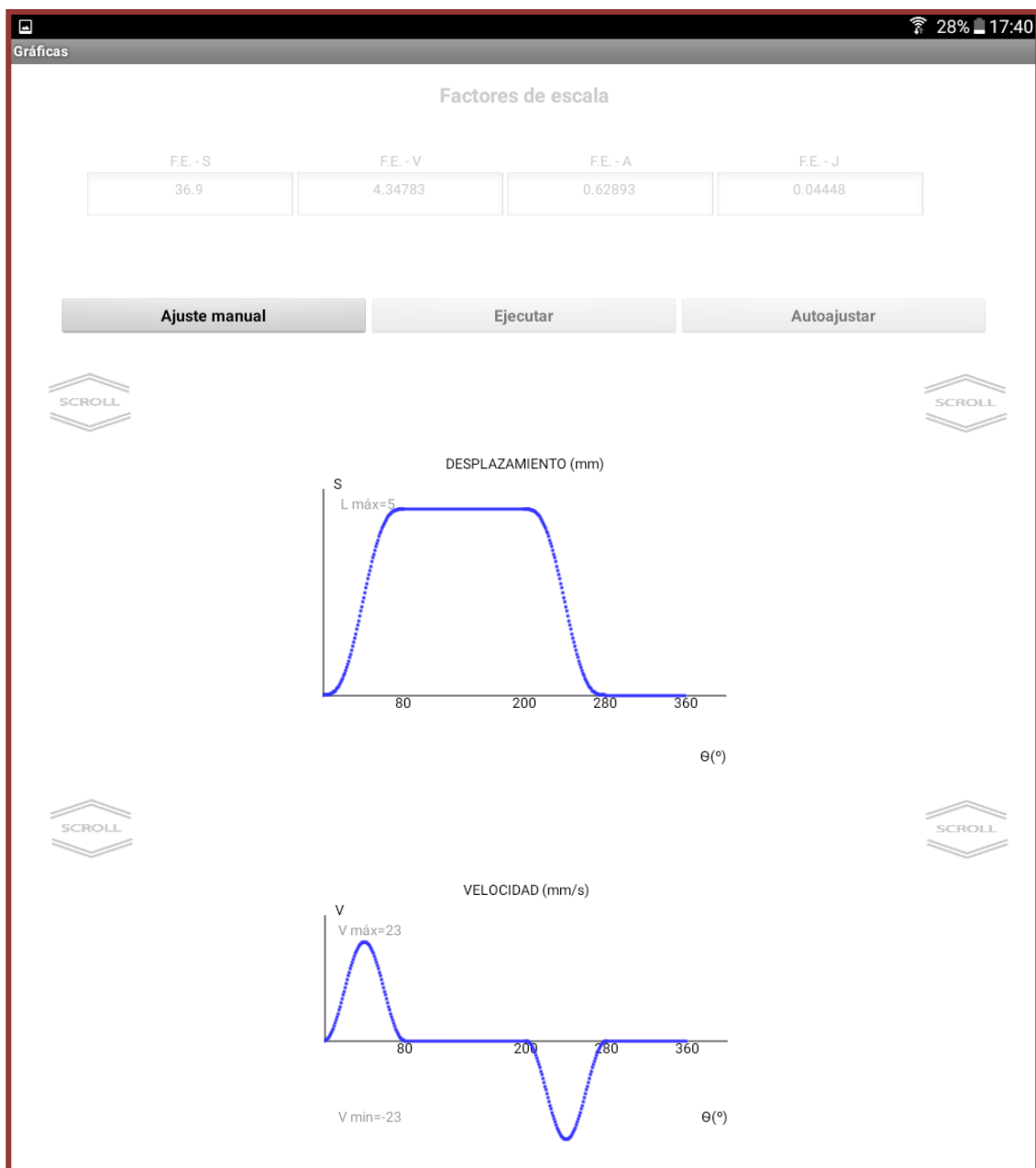


Figura 87. Pantalla “Gráficas”. 1ª Parte.

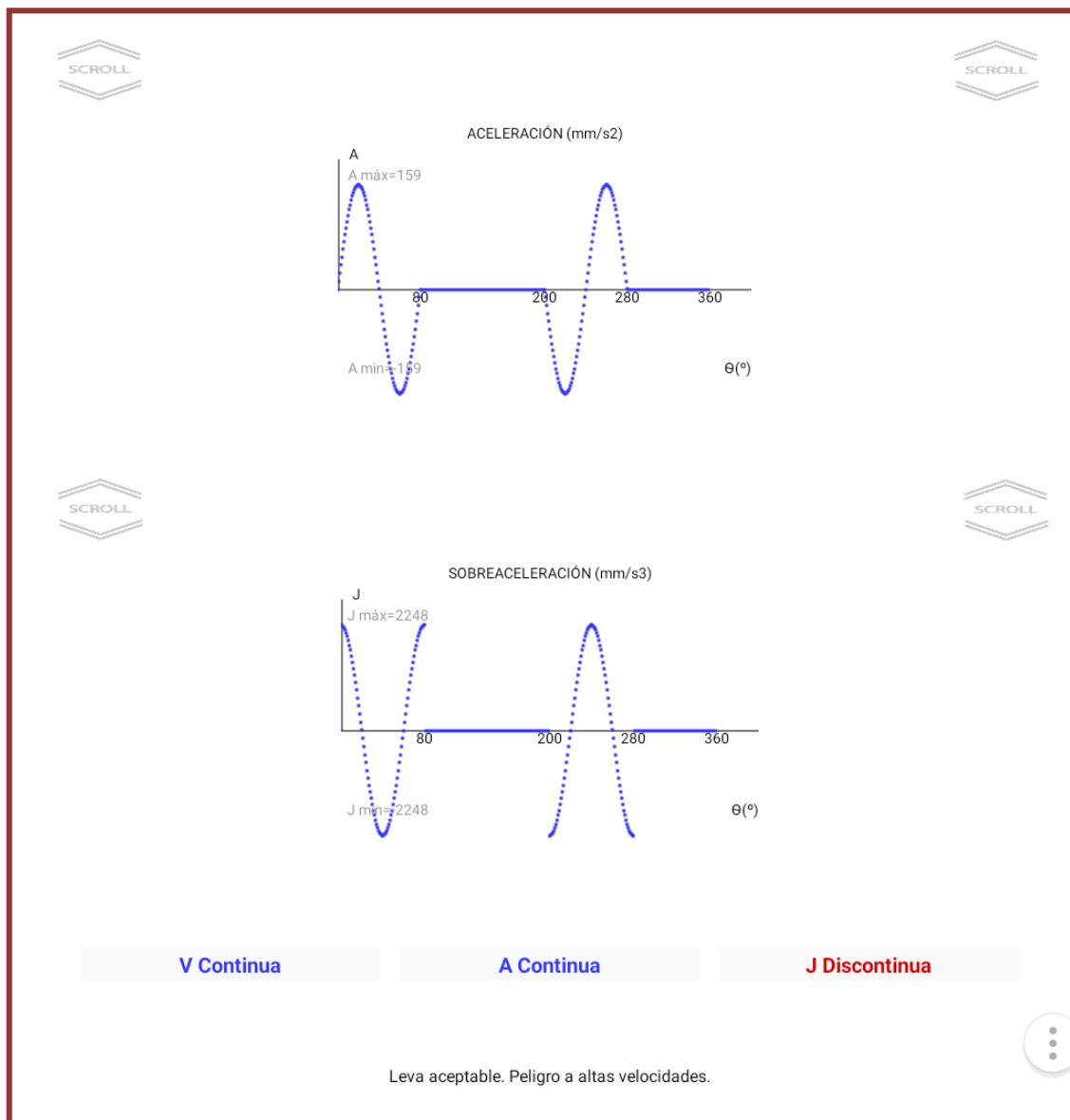


Figura 88. Pantalla “Gráficas”. 2ª Parte.

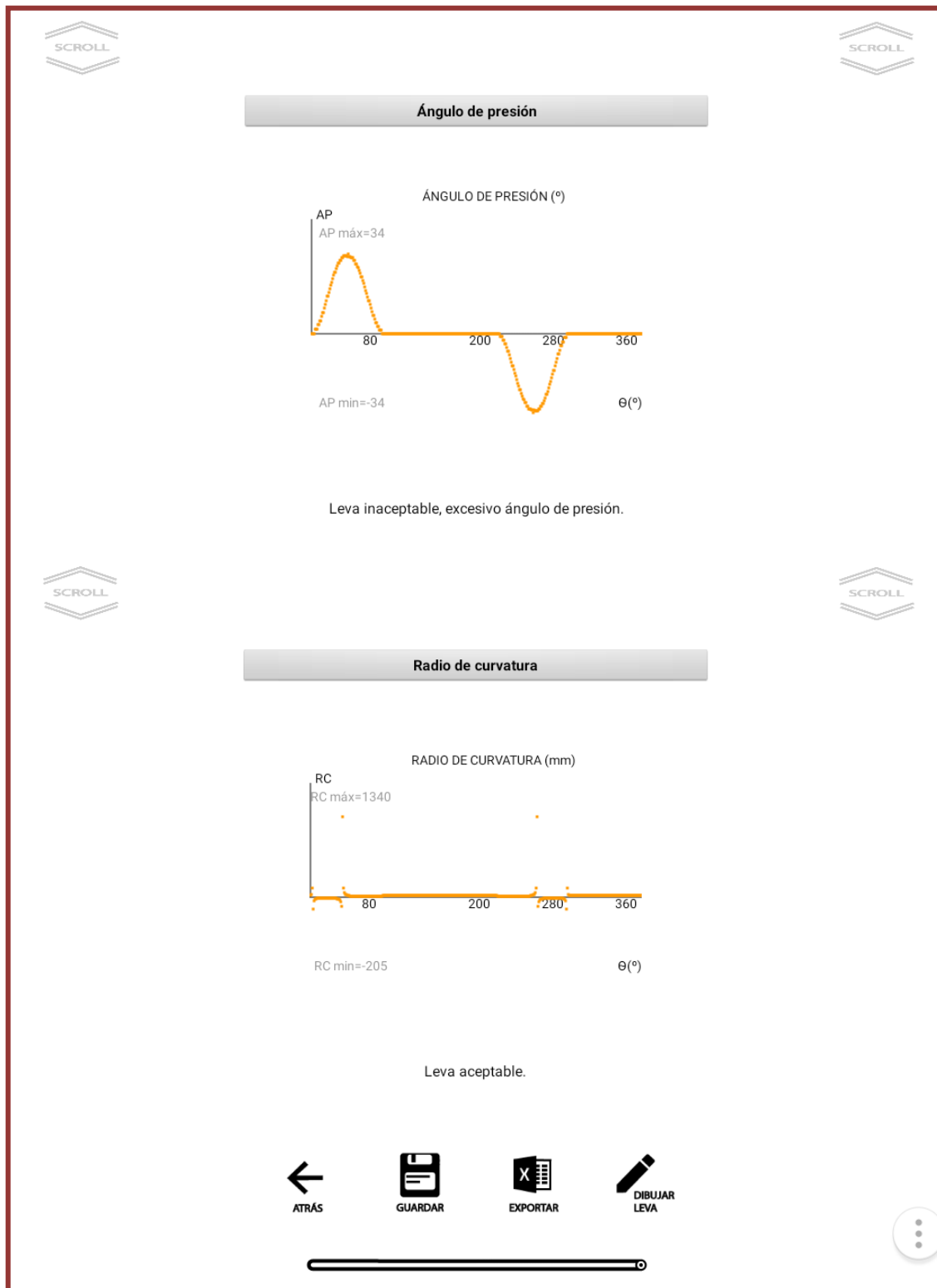


Figura 89. Pantalla "Gráficas". 3ª Parte.



Esta es sin duda la pantalla más importante de la aplicación. Como vemos hemos obtenido las gráficas cinemáticas que nos solicita el enunciado, de las cuales, tanto la de desplazamiento, velocidad como aceleración son continuas, lo que es requisito mínimo para poder aceptar nuestro mecanismo.

La gráfica de sobreaceleración vemos que presenta discontinuidades, pues como sabemos, los tramos cicloidales con los detenimientos no son compatibles en la tercera derivada. Aun así, estas discontinuidades se permiten, teniendo que llevar cuidado a altas velocidades de giro de la leva, como nos indica la app.

Más adelante, pulsando sobre los botones correspondientes vemos la representación y análisis del ángulo de presión y radio de curvatura. Por un lado, el radio de curvatura es aceptable, sin la generación de zonas cóncavas o convexas. Pero vemos que anteriormente, para los datos que nos proporciona el enunciado, la leva no es factible, pues tenemos un ángulo de presión de 34° , que supera el límite teórico visto de 30° para este tipo de seguidores traslacionales.

Por lo tanto, buscaremos una solución. Sabemos que para disminuir el ángulo de presión podemos aumentar el radio base, aumentar el radio del rodillo o modificar la excentricidad. Probaremos con aumentar los radios tanto base de la leva como del seguidor de rodillo. Introduciremos al programa $R_b=40$ mm. y $R_r=14$ mm.

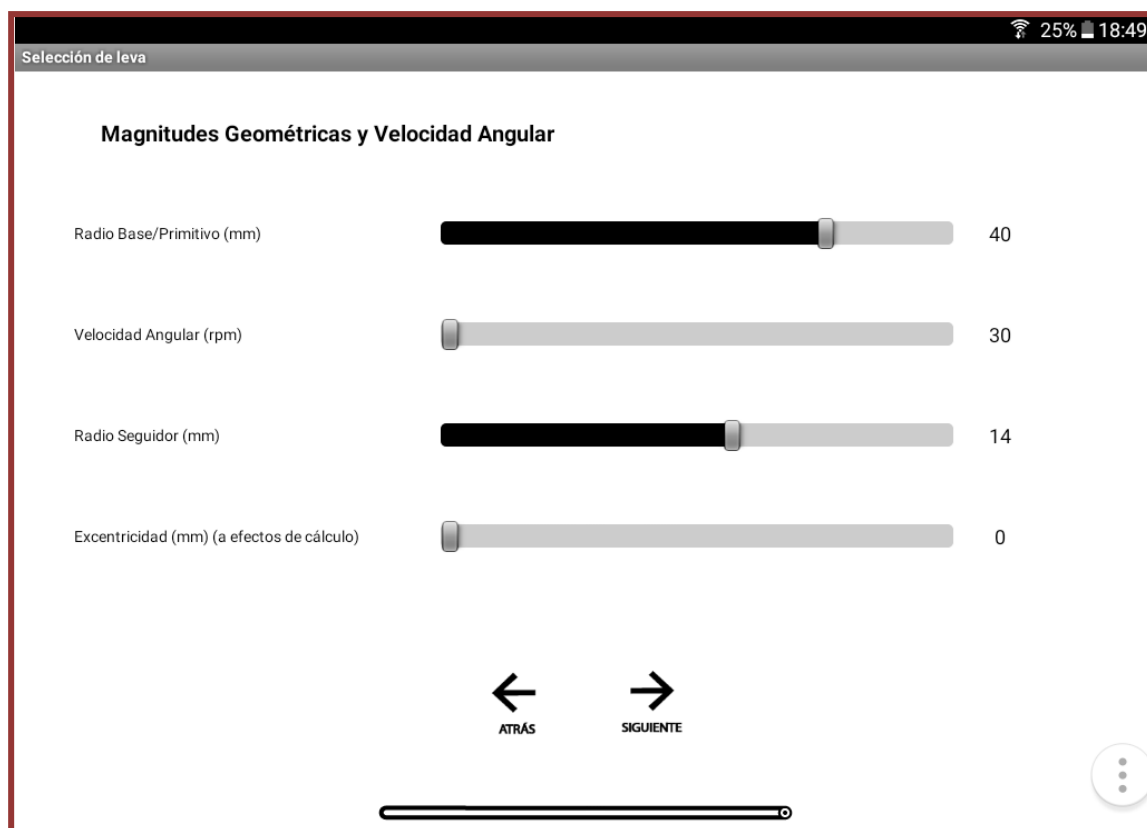


Figura 90. Reajuste magnitudes geométricas.



Al cambiar estos parámetros e introducir de nuevo las características de cada tramo volvemos a la pantalla “Gráficas” y comprobamos el cambio en el ángulo de presión y radio de curvatura. Los valores de las gráficas cinemáticas permanecerán intactos, pues estos cambios no afectan a las ecuaciones de movimiento.

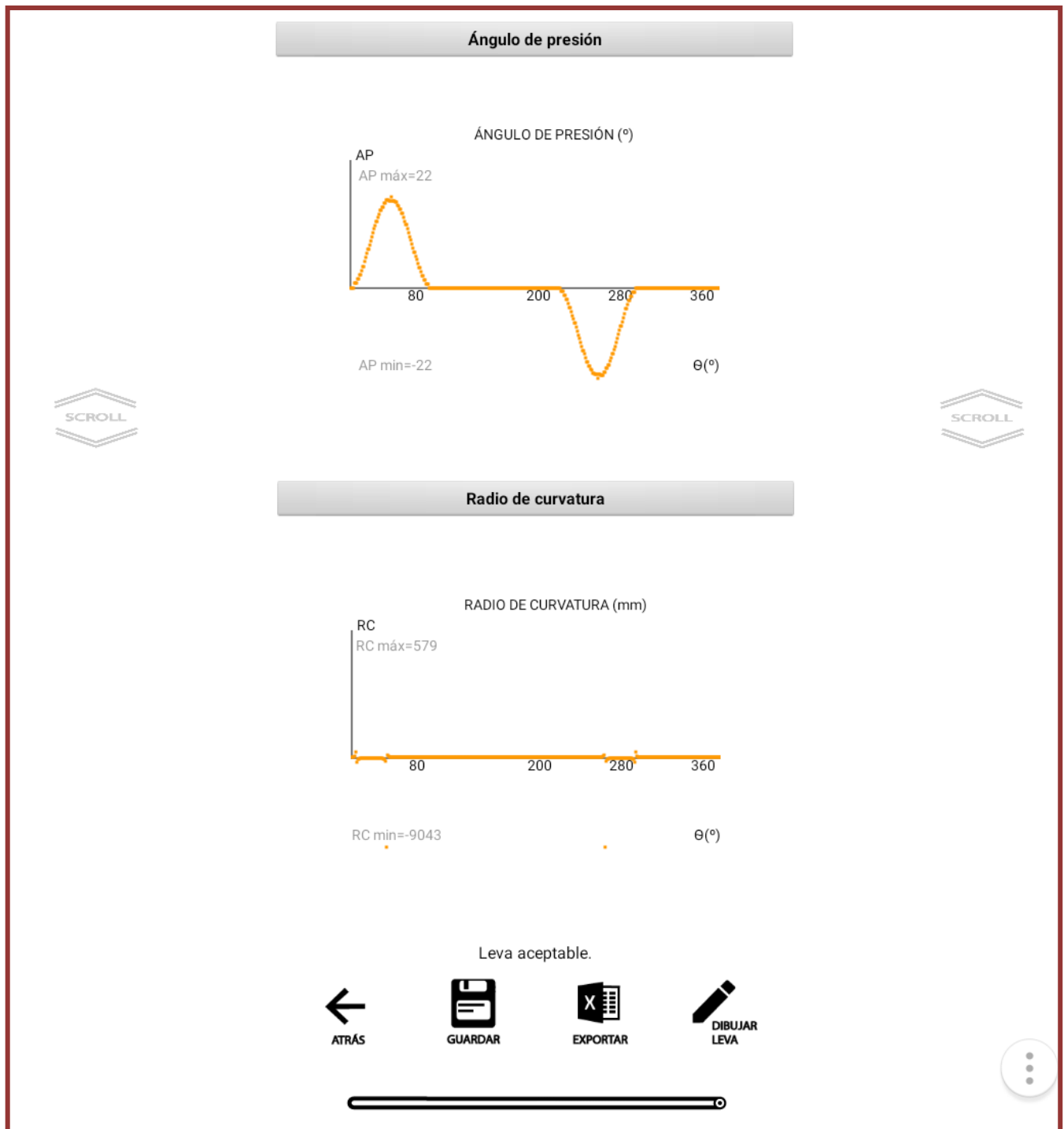


Figura 91. Modificación ángulo de presión y radio de curvatura.



Tras los cambios pertinentes comprobamos que el ángulo de presión ha disminuido, alcanzando valores máximos de tan sólo 22° , lo que permite la factibilidad del sistema. En cuanto a los valores de radio de curvatura, vemos que han variado pero siguen siendo aceptables. Con el diseño de una leva factible seguimos avanzando a la siguiente pantalla pulsando sobre “Dibujar Leva”, donde nos encontraremos con una representación del mecanismo.

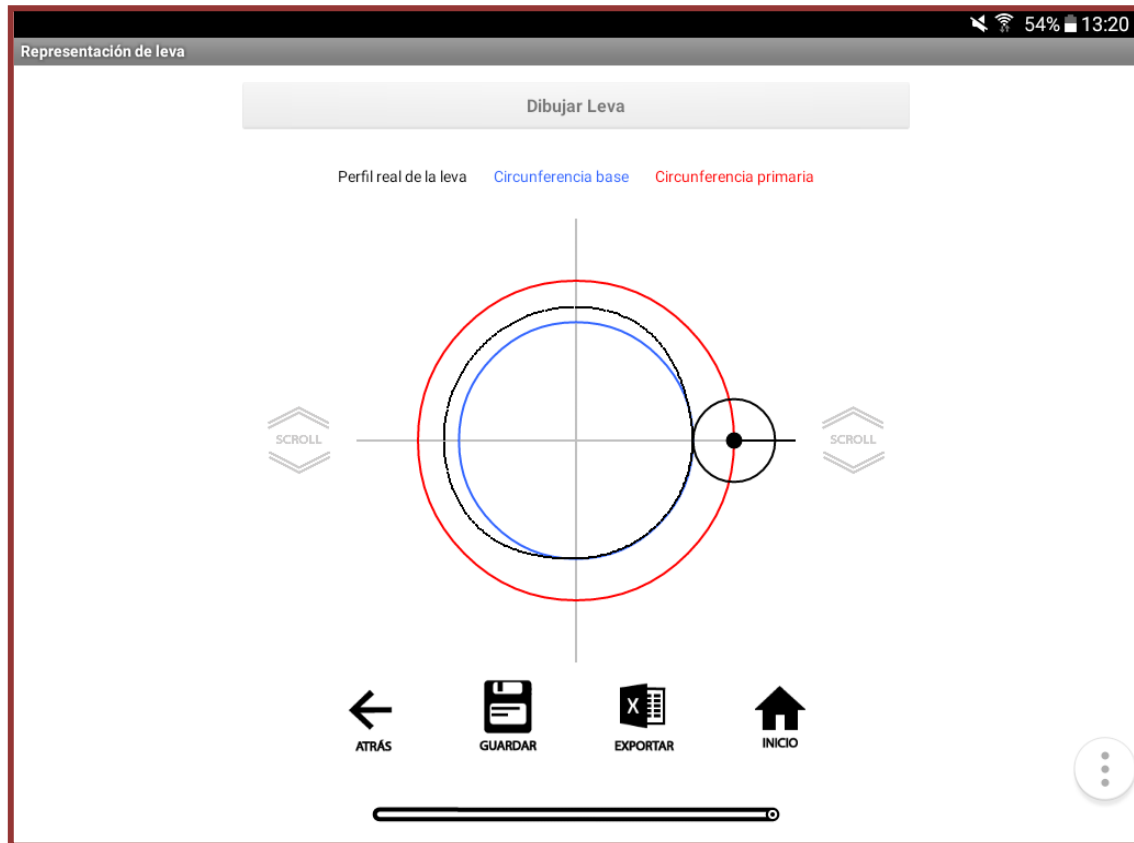


Figura 92. Pantalla “Representación de leva”.



Volviendo a la pantalla anterior, tenemos la posibilidad de guardar una imagen en formato *.png* de todas las representaciones de la pantalla “Gráficas” pulsando sobre el botón “Guardar”.

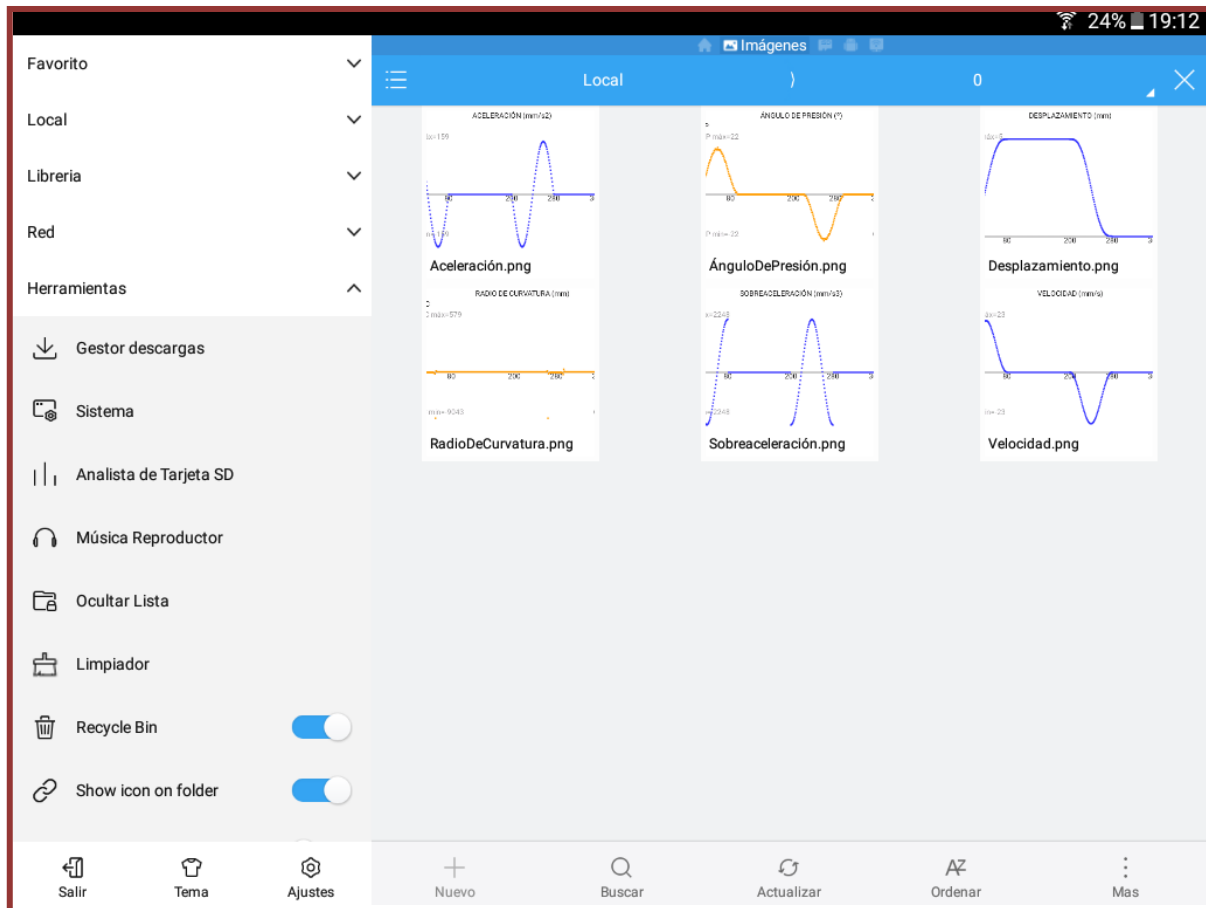


Figura 93. Guardado de gráficas.



Por otro lado, pulsando sobre el botón “Exportar”, se guardará el archivo “datos.csv” en la carpeta de descargas del dispositivo. Utilizaremos este archivo para obtener las gráficas en el software Microsoft Excel.

Aprovecharemos la opción que nos ofrece la aplicación para compartir estos archivos con nosotros mismos utilizando Gmail. De esta manera podremos acceder a los correos desde cualquier ordenador con acceso a internet.

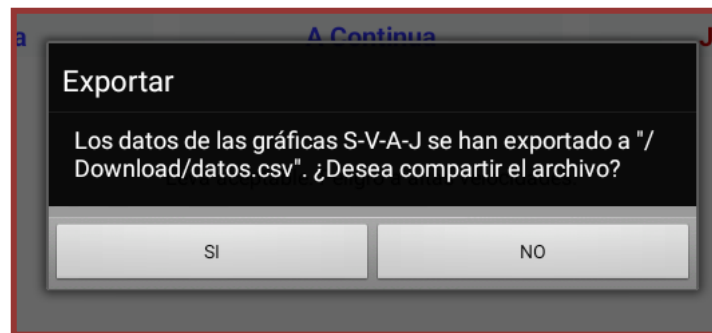


Figura 94. Exportar “datos.csv”.

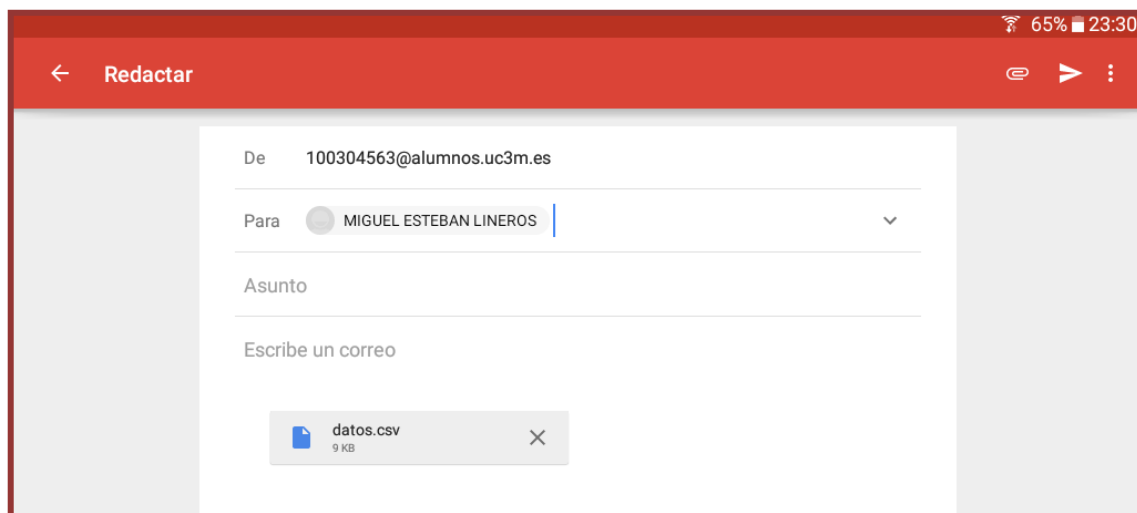


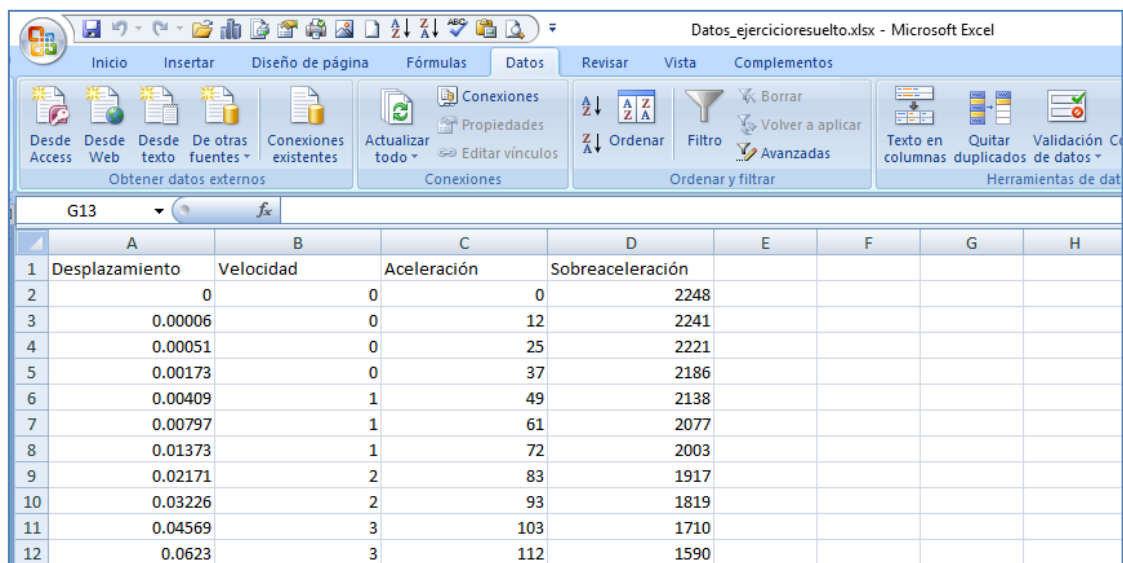
Figura 95. Compartir “datos.csv” vía Gmail.



Ambos archivos deben contener sus datos entrecomillados en una sola columna y separados por una coma. Para tratar correctamente los datos de los dos archivos exportados debemos abrirlos en el software Excel y proceder de la siguiente manera con los datos marcados en el programa:

Datos >> Texto en columnas >> Delimitados >> Separadores/Coma >> Finalizar

De este modo visualizaremos de forma correcta los números en Excel. Del archivo “datos.csv” extraemos los puntos de las gráficas cinemáticas. Mediante el manejo del software podemos sacar también las representaciones a nuestro gusto.



	A	B	C	D	E	F	G	H
1	Desplazamiento	Velocidad	Aceleración	Sobreaceleración				
2	0	0	0	2248				
3	0.00006	0	12	2241				
4	0.00051	0	25	2221				
5	0.00173	0	37	2186				
6	0.00409	1	49	2138				
7	0.00797	1	61	2077				
8	0.01373	1	72	2003				
9	0.02171	2	83	1917				
10	0.03226	2	93	1819				
11	0.04569	3	103	1710				
12	0.0623	3	112	1590				

Figura 96. Detalle fichero “datos” en Excel.

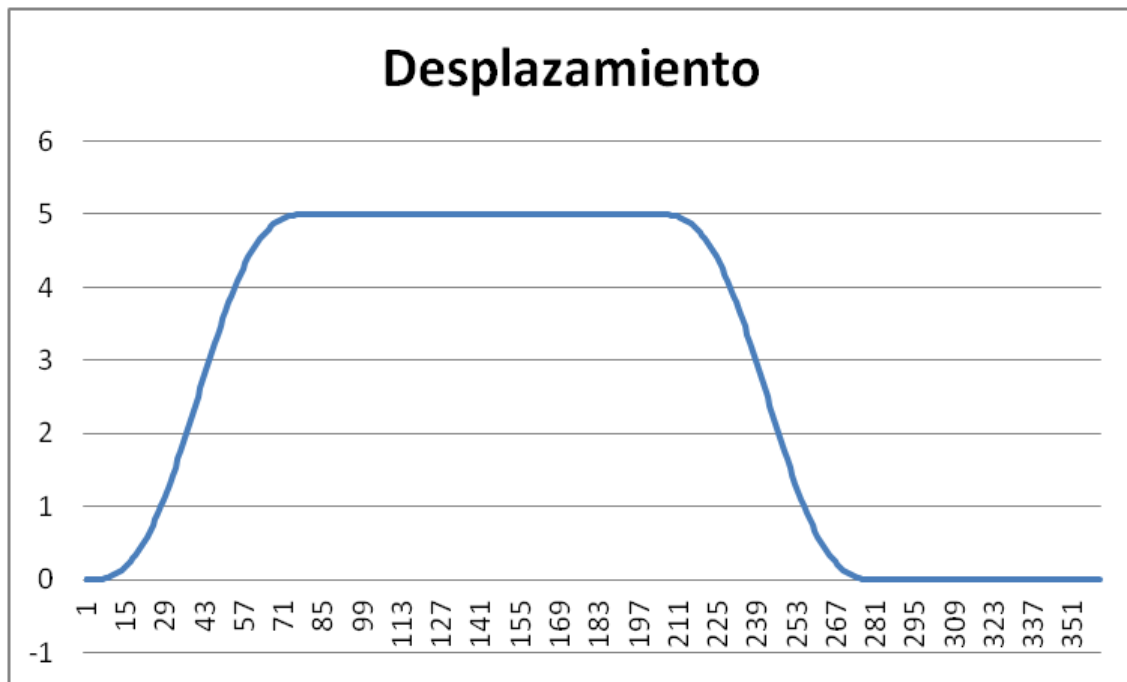


Figura 97. Gráfica de desplazamiento en Excel.

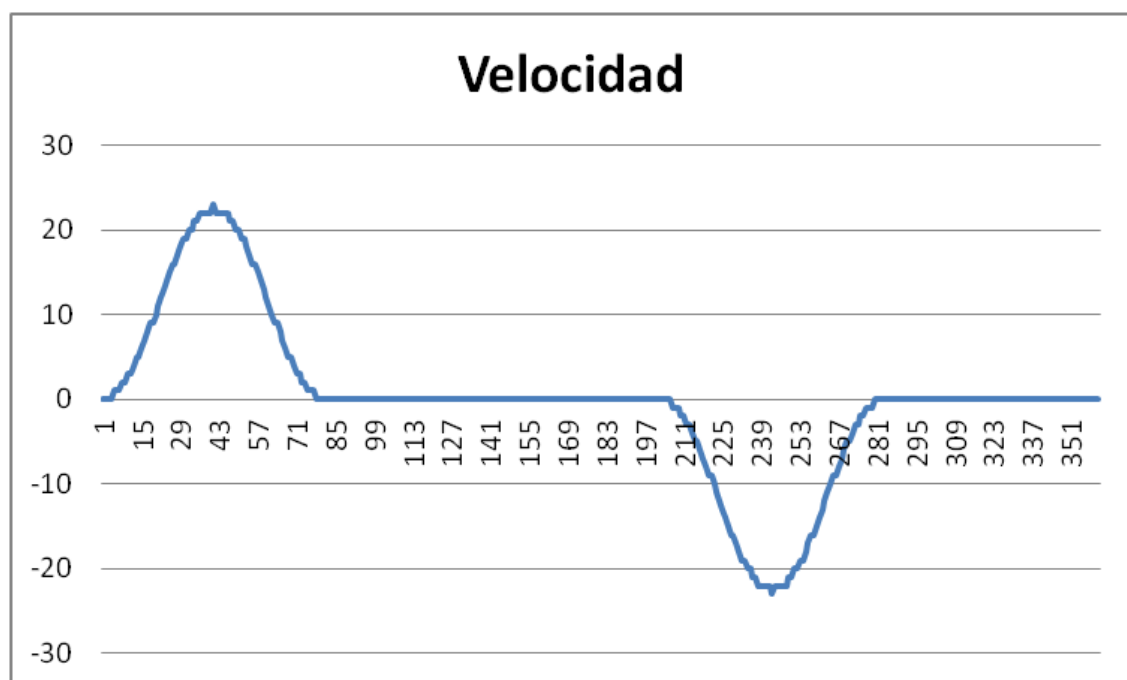


Figura 98. Gráfica de velocidad en Excel.

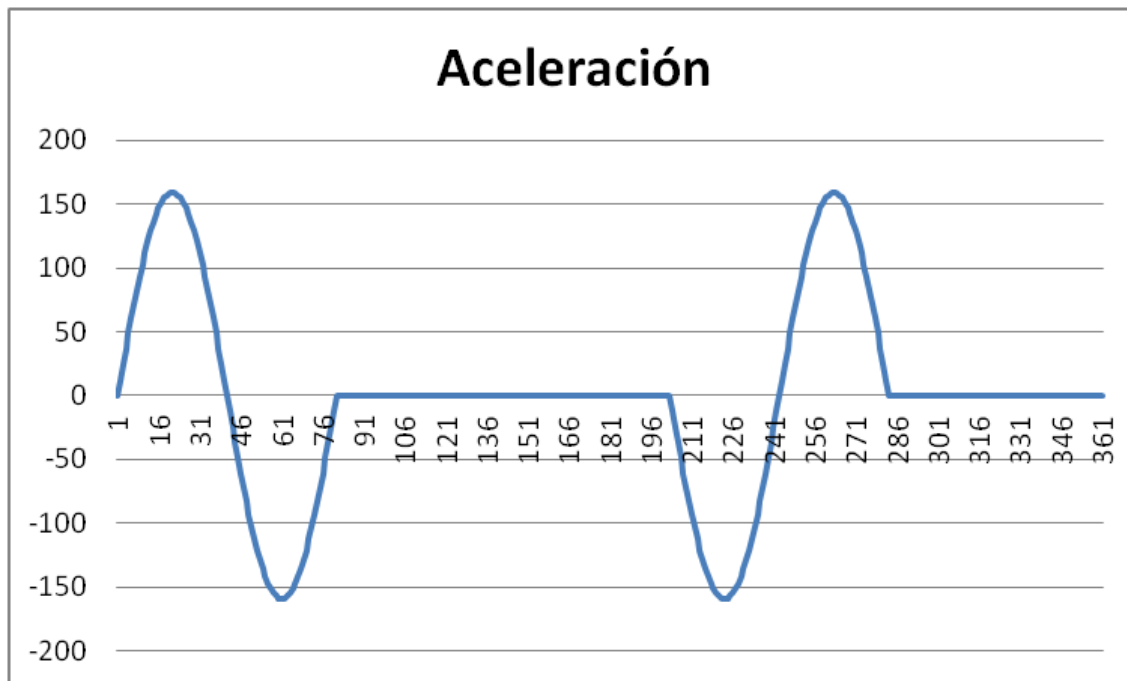


Figura 99. Gráfica de aceleración en Excel.

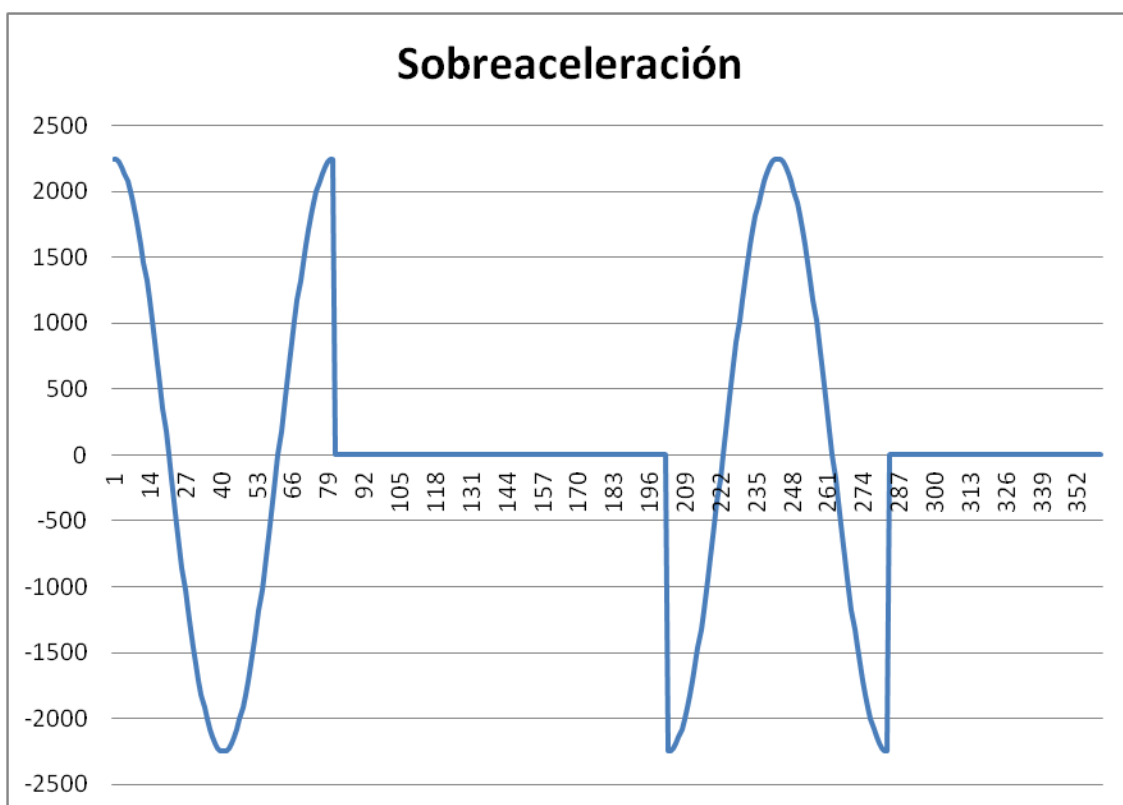


Figura 100. Gráfica de sobreaceleración en Excel.



Por otro lado, como hemos comentado anteriormente, desde la pantalla “Representación de leva” exportaremos al dispositivo el archivo “puntosleva.csv” de donde extraemos los puntos del contorno de la leva en coordenadas cartesianas. Así podremos representar la leva en Excel. De la misma manera, compartiremos dicho documento mediante otra aplicación externa ya que la aplicación “Cálculo de levass” nos da la opción de hacerlo.

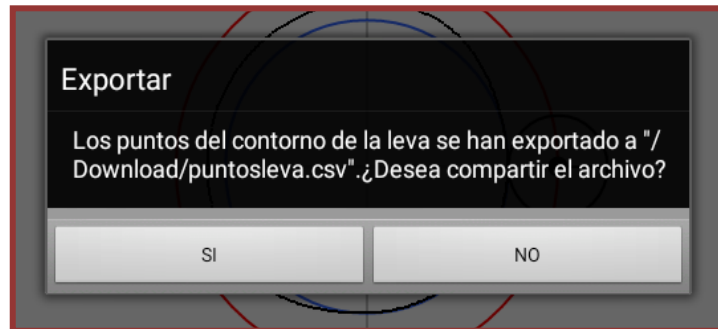


Figura 101. Exportar “puntosleva.csv”.

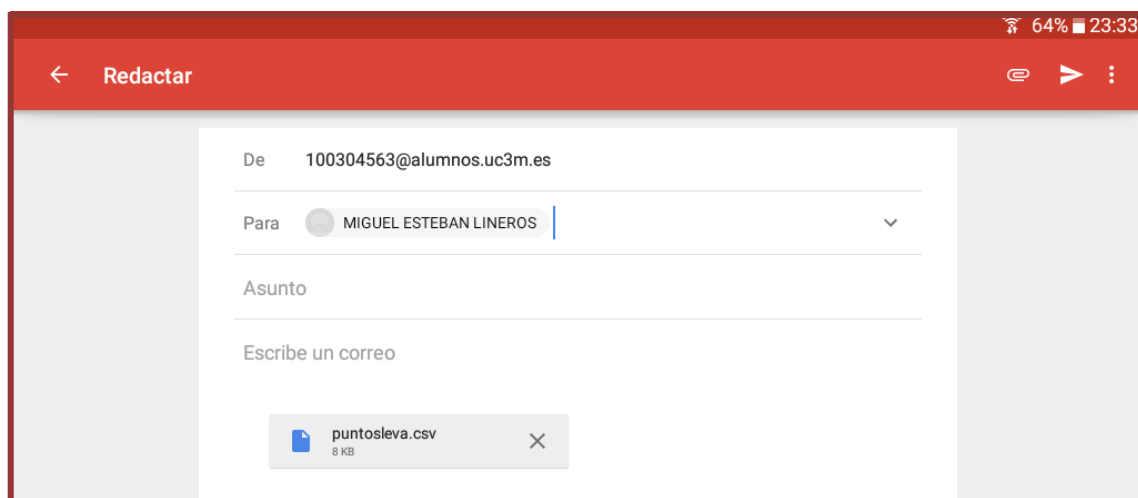


Figura 102. Compartir “puntosleva.csv” vía Gmail.

PuntosLeva_ejercicioresuelto.xlsx - Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	Coord. X	Coord. Y							
2	40	0							
3	39.99397	-0.6981							
4	39.97615	-1.396							
5	39.94691	-2.09353							
6	39.90664	-2.79054							
7	39.85573	-3.48692							
8	39.79453	-4.18257							
9	39.72339	-4.87742							
10	39.64266	-5.57141							
11	39.55266	-6.26453							
12	39.45367	-6.95675							
13	39.34596	-7.64808							
14	39.22979	-8.33855							
15	39.10536	-9.02818							
16	38.97286	-9.71703							

Figura 103. Detalle fichero “puntosleva” en Excel.

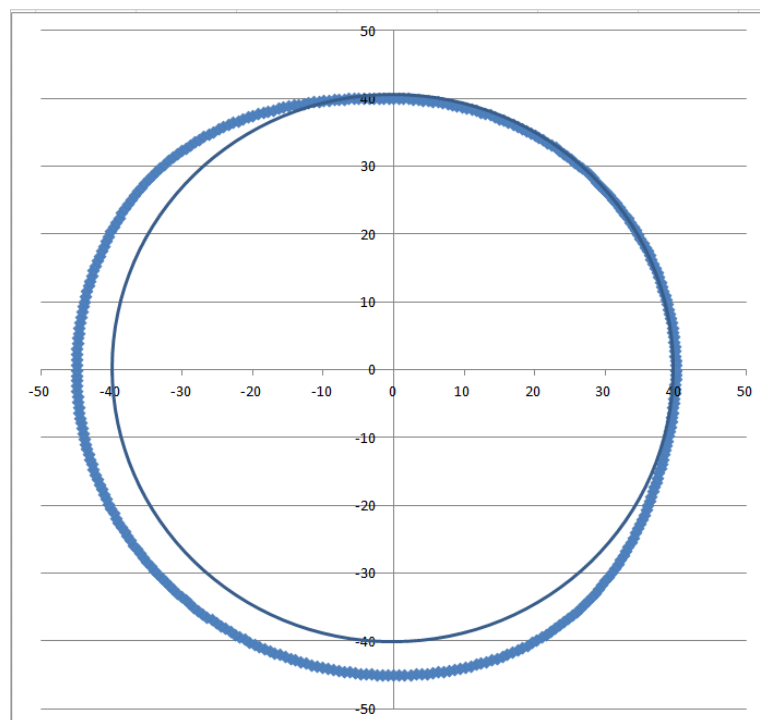


Figura 104. Leva representada en Excel.

Situando sobre la gráfica de Excel una circunferencia de radio 40 (nuestro radio base), somos capaces de identificar mejor el perfil de la leva.



Poder dotar a la app “Cálculo de levass” con la propiedad de exportar archivos .csv y compartirlos es toda una ventaja que el usuario puede aprovechar. Disponer de estos archivos en Excel hace que el trabajo de análisis sea más minucioso y detallado. Además, puesto que este software de hojas de cálculo es internacional, se puede dar mucho uso a los archivos exportados.

Una buena prueba de ello, que durante el desarrollo de este Trabajo Fin de Grado se ha tratado de aplicar, es el uso del archivo “puntosleva.csv” para imprimir levass en 3D. Para hacer esto, se ha utilizado el software de diseño CAD, Autodesk Inventor, que permite importar hojas de cálculo con coordenadas cartesianas de puntos. De esta forma se ha hecho un modelo 3D de la leva resuelta en este ejercicio a través de las sencillas funciones del software de diseño empleado.

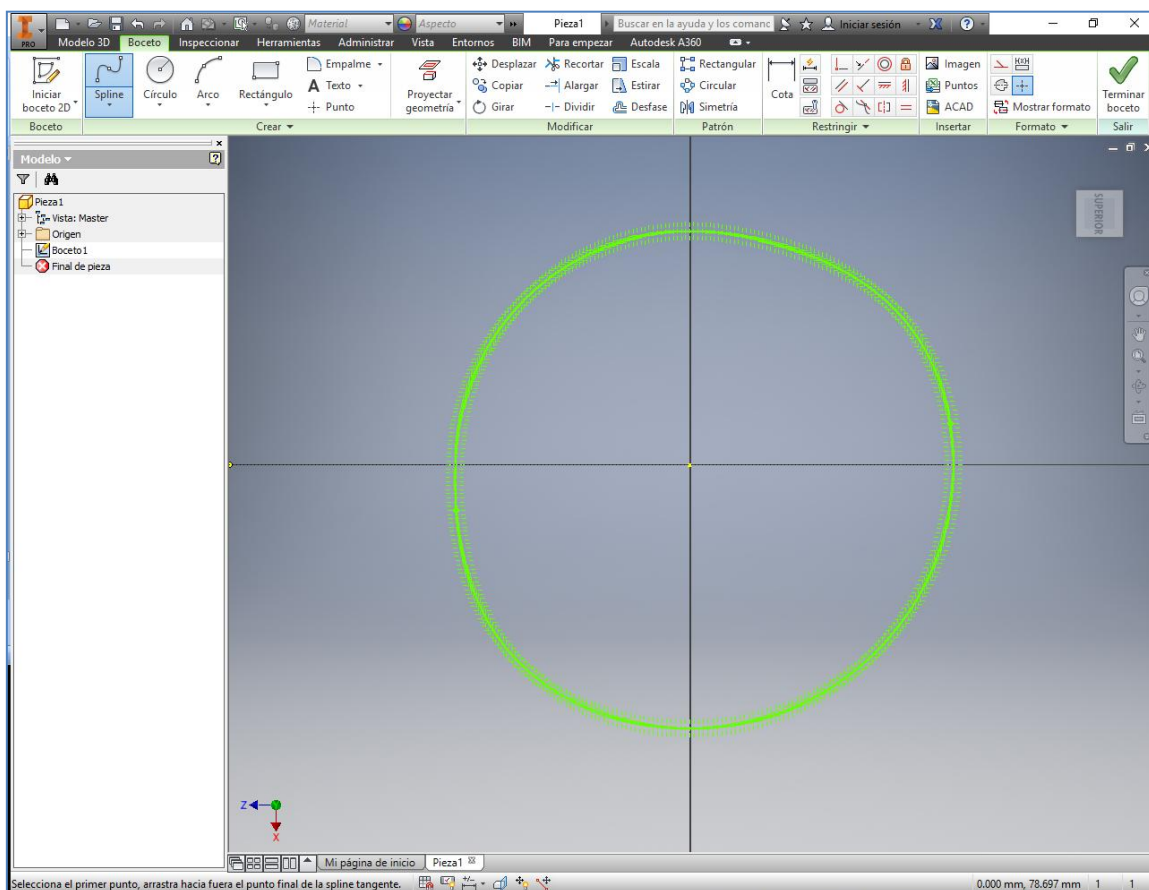


Figura 105. Diseño de leva en Autodesk Inventor.

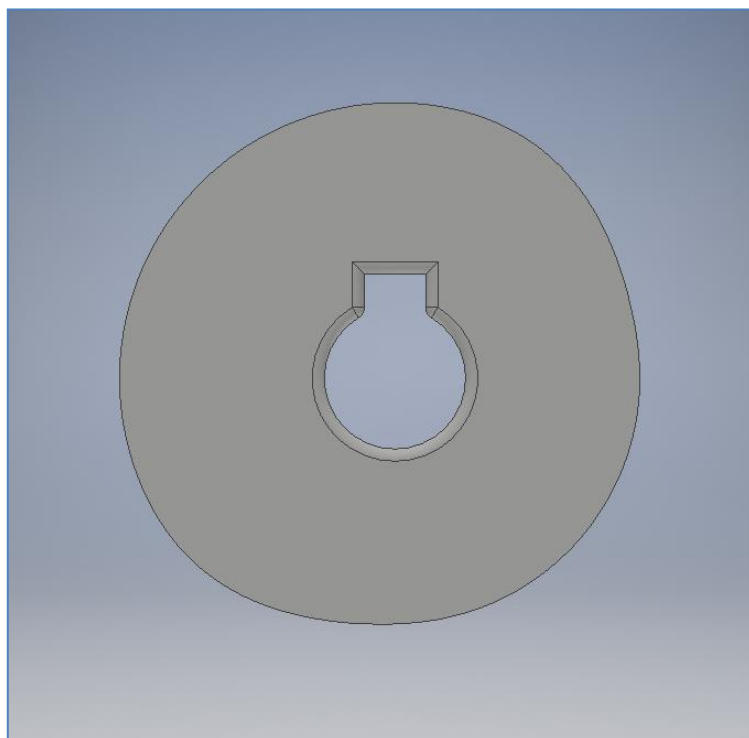


Figura 106. Modelo CAD planta.

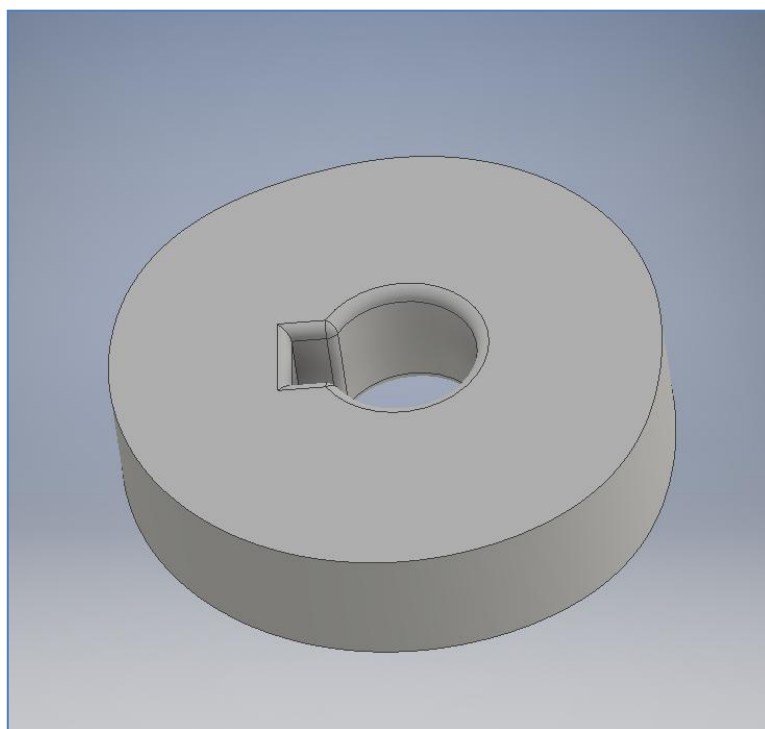


Figura 107. Modelo CAD.



Una vez hecho el modelo de la leva con las dimensiones deseadas, necesitamos el archivo en extensión *.stl*, un formato donde únicamente se definirá a geometría de la pieza, sin tener en cuenta información como color, texturas o propiedades físicas. Para la impresión necesitaremos una impresora 3D y un ordenador con cualquier software conectado a la impresora capaz de importar modelos CAD en la extensión apropiada. Para este caso utilizaremos el software libre Repetier.

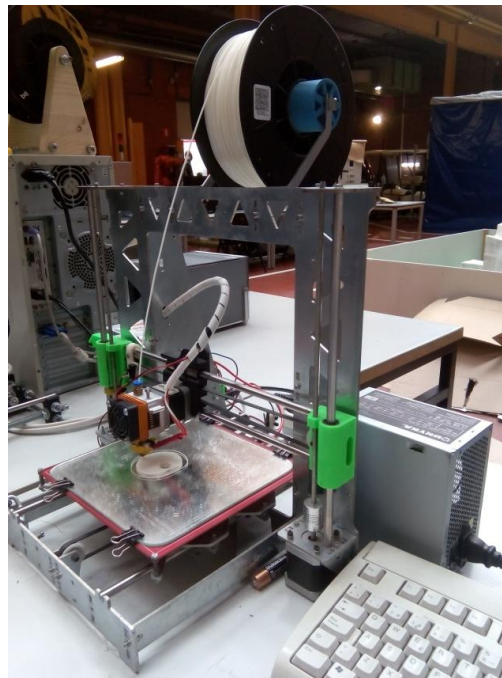


Figura 108. Impresión 3D.

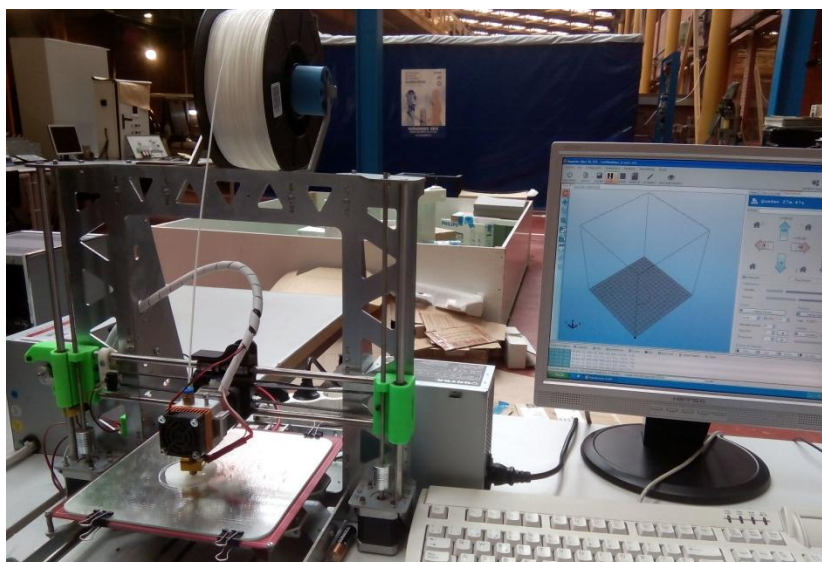


Figura 109. Equipo para impresión 3D.

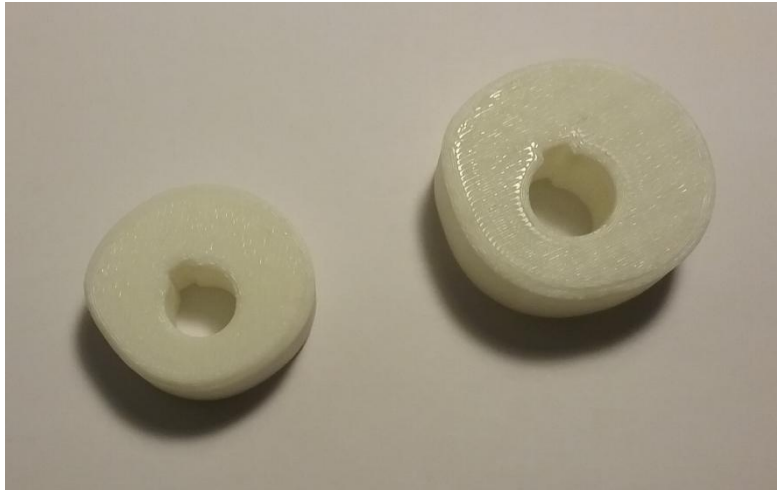


Figura 110. Levas en 3D de diferentes tamaños.



5.2. VERIFICACIÓN EN SC LEVAS

Para mostrar el correcto funcionamiento de la app “Cálculo de levas”, se ha realizado el mismo ejercicio en el software tomado como referencia y anteriormente comentado SC Levas. Con él comprobaremos que las curvas cinemáticas adoptan los valores adecuados en nuestra aplicación y verificaremos el perfil de la leva diseñada.

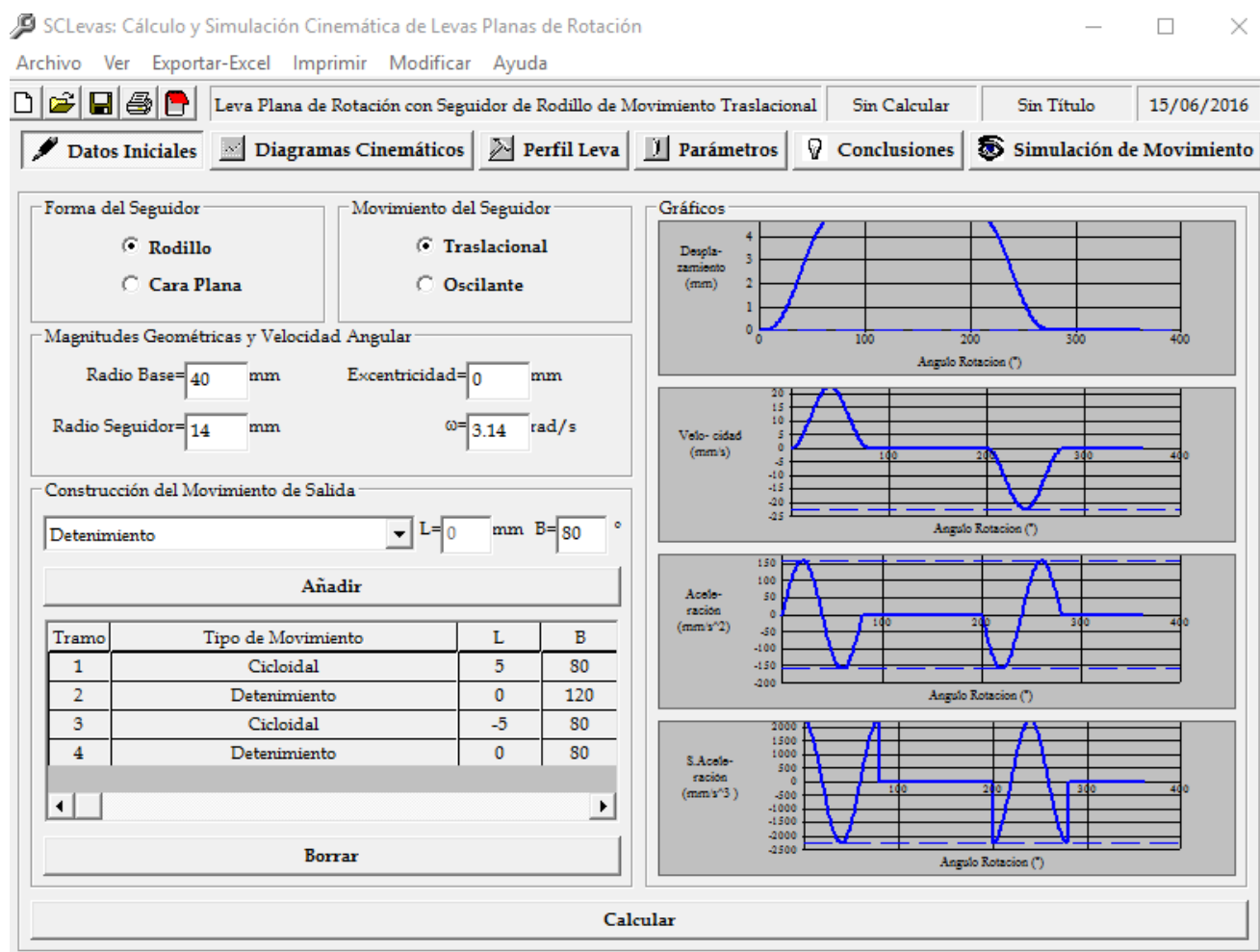


Figura 111. Configuración sistema leva-seguidor en SC Levas.

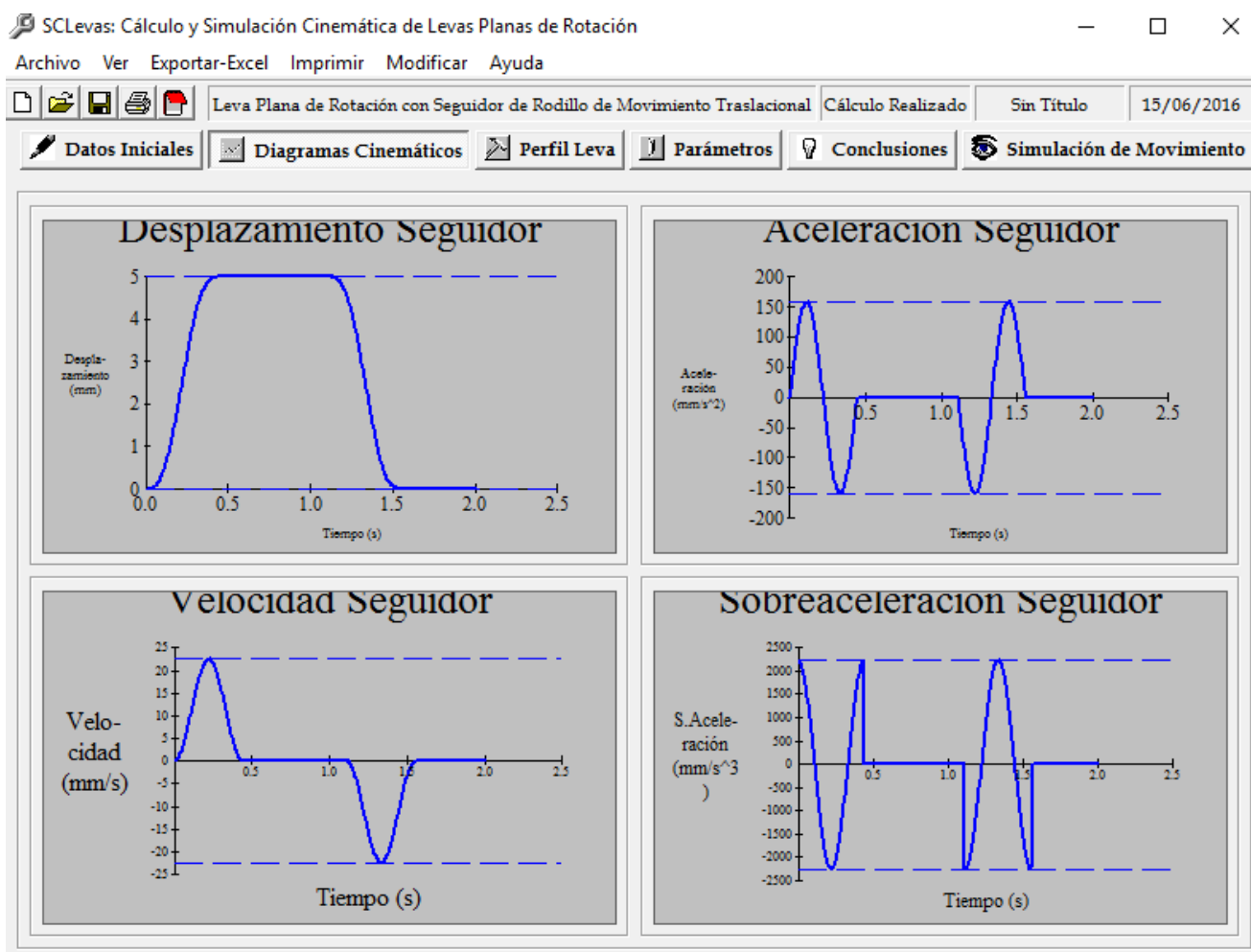


Figura 112. Diagramas cinemáticos en SC Levas.

Se puede apreciar como los valores máximos que se alcanzan en la gráficas cinemáticas resultado de introducir los parámetros en SC Levas coinciden con los obtenidos en la aplicación informática desarrollada.

De la misma forma, visualizaremos cómo en el perfil real de la leva no existe diferencia entre una y otra herramienta.

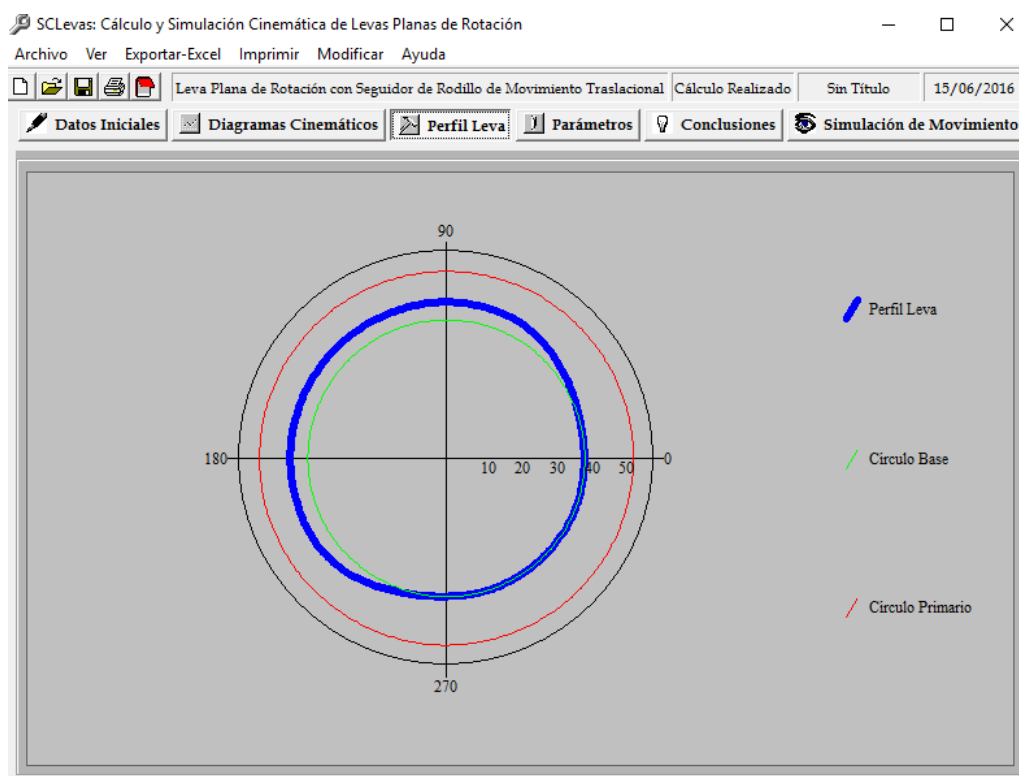


Figura 113. Perfil de la leva en SC Levas.

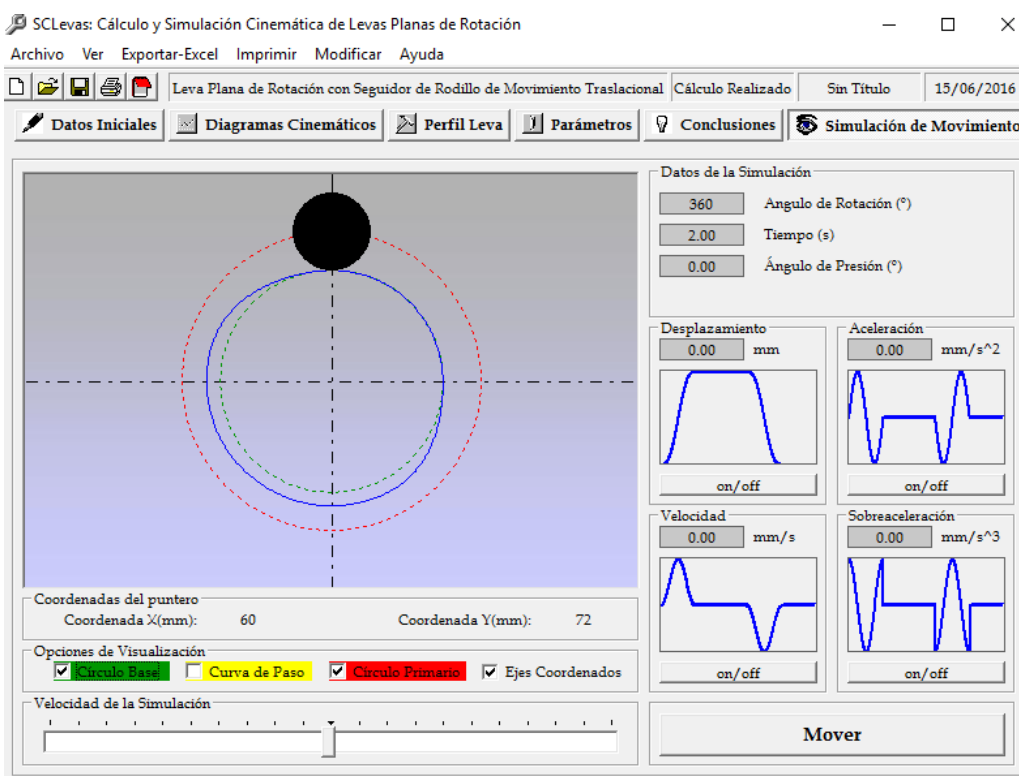


Figura 114. Mecanismo leva-seguidor en SC Levas.



6. CONCLUSIONES Y TRABAJO FUTURO





6.1. CONCLUSIONES

Llegados a este punto del Trabajo Fin de Grado podemos afirmar que el objetivo principal del mismo, *“diseño e implementación de una app (aplicación informática multiplataforma) para el cálculo de levas planas rotatorias con seguidor traslacional”*, se ha cumplido satisfactoriamente.

Para llegar a completar nuestra meta se han cumplido, de forma metódica, los objetivos secundarios detallados al comienzo de este trabajo. Además, se ha intentado que la app fuese lo más didáctica y sencilla de uso posible para hacer de ella una herramienta de utilidad.

De esta manera, se implementó una estructura básica de la app “Cálculo de levas” que se ha ido ampliando y modificando conforme iban surgiendo problemas y se iban cumpliendo los objetivos, durante el desarrollo de la aplicación.

La entrada de parámetros en la aplicación informática, como son el tipo de seguidor, número de tramos, velocidad angular y magnitudes geométricas, junto con la posterior confección de los tramos, permite al usuario la visualización y análisis de las gráficas cinemáticas y de análisis de la leva diseñada así como una representación del mecanismo.

La app “Cálculo de levas” trata de ir más allá, dando la opción al usuario de la app de exportar, guardar y compartir los archivos generados, tras la implementación del sistema leva-seguidor. Todo ello de forma didáctica, original y sencilla.

Mediante la metodología seguida se ha conseguido explotar todo el potencial del software utilizado, ApplInventor, minimizando sus limitaciones. Además, no sólo se ha buscado un resultado funcional y útil de la aplicación informática sino llegar a ello de una forma estética y con sentido de diseño gráfico, lo que potencia la originalidad del proyecto.

Para demostrar con integridad el funcionamiento y alcance de la aplicación, en el documento se ofrecen una serie de juegos de prueba donde se plantean distintas situaciones que el usuario puede encontrar en la app “Cálculo de levas”. Se ha comprobado cómo gracias a las distintas herramientas previstas en el software elegido se pueden tratar los resultados de un problema genérico planteado acudiendo a programas externos, en este caso a Excel.

La intercomunicación de ApplInventor con otros programas dota a la app “Cálculo de levas” de una gran ventaja sobre sus competidores en el mercado Android. Por ejemplo, una aportación especial es la posibilidad de exportar los resultados obtenidos en la aplicación a entornos que permiten la impresión 3D de las levas diseñadas por el propio usuario, de forma rápida y fácil.

Por último, a medida que se ha ido avanzando con el proyecto se ha podido comprobar que el uso de aplicaciones multiplataforma para la realización de cálculos tecnológicos es muy provechoso y, seguramente, cada vez sea más utilizado en todos los sectores de la industria. Los cálculos técnicos adquieren con este tipo de herramientas una mayor eficiencia de uso, cercanía



al usuario, tiempos menores de trabajo y, por supuesto, ahorro económico, todo ello desde un soporte completamente portátil y manejable como un Smartphone o una tablet.

La app “Cálculo de levas” ha sido probada con éxito en varias plataformas, entre otras, en los siguientes dispositivos:

- Smartphone **Samsung S4 mini**; v. 4.4.4 de Android
- Smartphone **Samsung Galaxy Core**; v. 4.1.2 de Android
- Smartphone **Samsung Galaxy S4**; v. 4.4 de Android
- Smartphone **bq Aquaris E4.5**; v. 4.4.2 de Android
- Smartphone **HTC one M9**; v. 6.0 de Android
- Tablet **Samsung Galaxy Tab A**; v. 5.0.2 de Android
- Tablet **bq Edison 3**; v. 4.4.2 de Android

6.2. TRABAJOS FUTUROS

Al tratarse de un producto nuevo e innovador las posibilidades de expansión y mejora son amplias. Entre otras, como líneas de mejora se proponen las siguientes:

- Incorporación de otros idiomas.
- Investigar sobre otros sistemas operativos y realizar el desarrollo de la app en iOS y Windows Phone, como principales sistemas operativos del mercado después de Android.
- Desarrollo en otro entorno de programación, AndroidStudio o Eclipse, para suprimir errores a través de la implementación de hilos de ejecución. Permitir de esta manera mejorar la visualización de las gráficas, pudiendo realizar scroll sobre ellas e incluso zoom.
- Ampliar número de tramos y tipo de curvas a utilizar en el diseño de levas.
- Publicar aplicación en PlayStore.
- Extrapolación de la metodología de diseño de la app a otros elementos mecánicos o combinación de ellos.
- Mejorar el aspecto: Estética 100% Material Design, adaptándonos a los estándares de Google para dar así un aspecto más global a la app “Cálculo de levas”.
- Potenciar la función de exportar, incorporando los datos de ángulo de presión y radio de curvatura. Estudiar la posibilidad de utilizar “La Nube” para guardar los resultados



obtenidos, permitiendo de esta forma la creación de una base de datos que almacene las imágenes guardadas por el usuario.

Dichas mejoras dotarían a la app “Cálculo de levass” con una versatilidad y funcionalidad excelentes, pudiéndose convertir en una herramienta útil y de referencia en el mercado.





7. PRESUPUESTO





El desarrollo de este proyecto conlleva una serie de gastos económicos estimados que se desarrollan a continuación. El desglose de los costes se ha dividido en cuatro grupos: formación, desarrollo, equipo informático y otros costos.

- **Formación:** Estimación del coste necesario para el aprendizaje del lenguaje de programación utilizado en AppInventor y de los fundamentos de Android, así como programas de diseño gráfico para potenciar la estética final de la aplicación. Se creará una pequeña aplicación de prueba para explorar los fundamentos del software elegido y familiarizarnos con el entorno de programación.
- **Desarrollo:** Coste imputable al tiempo desempeñado para la creación y puesta de funcionamiento de la app “Cálculo de levass” por parte del autor del proyecto, Miguel Esteban Lineros, como ingeniero.
- **Informática:** Se estima el coste asociado al uso del Pc de sobremesa como del software empleado y los dispositivos requeridos.
- **Otros Costes:** Aglomera el resto de costes derivados del proyecto como impresión de documentos, desplazamientos...

<u>Formación</u>	<u>Horas</u>	<u>Coste (€)/hora</u>	<u>Coste imputable (€)</u>
Lenguaje de programación + Pruebas	30	15	450
Android y Google Design + Pruebas	20	15	300
Adobe Illustrator + Pruebas	20	15	300
Total formación			1050

Tabla 1. Costes de formación.

<u>Desarrollo</u>	<u>Horas</u>	<u>Coste (€)/hora</u>	<u>Coste imputable (€)</u>
Programación de “Cálculo de levass”	180	25	4500
Documentación y redacción de la memoria	40	20	800
Total desarrollo			5300

Tabla 2. Costes de desarrollo.



<u>Informática</u>	<u>Coste (€)</u>	<u>Amortización</u>	<u>Coste imputable (€)</u>
Pc + Office 2010 + Conexión Internet	1600	1/8	200
Smartphone	200	1/20	10
Tablet	250	1/10	25
ApplInventor	0	-	0
Complementos para ApplInventor	0	-	0
Adobe Illustrator	200	1/2	100
Total informática			335

Tabla 3. Costes de informática.

<u>Otros Costes</u>	<u>Coste imputable (€)</u>
Desplazamientos	200
Impresión de documentos y material de oficina	240
Total Otros Costes	440

Tabla 4. Otros costes.

<u>Resumen de costes</u>	<u>Coste imputable (€)</u>
Formación	1050
Desarrollo	5300
Informática	335
Otros Costes	440
Total Proyecto Provisional	7125

Tabla 5. Resumen de costes.

<u>Presupuesto final</u>	<u>Coste imputable (€)</u>
Gastos indirectos y beneficios adicionales + 20%	1425
Total Proyecto (Sin impuestos)	8550

Tabla 6. Presupuesto final.



8. BIBLIOGRAFÍA





8.1. REFERENCIAS BIBLIOGRÁFICAS CONSULTADAS

- [1] Norton, R. L. (2009), "Diseño de maquinaria." Síntesis y análisis de máquinas y mecanismos. Cuarta edición. ISBN: 978-970-106884-7. Ed.: Mc Graw Hill.
- [2] Norton, R. L. (2002), "Cam Design and Manufacturing Handbook." Industrial Press, New York.
- [3] Rothbart, H. A. (2003), "Cam Design Handbook." Ed.: Mc Graw Hill.
- [4] Norton, R. L. (1998), "Effect of Manufacturing Method on Dynamic Performance of Cams." Mechanism and Machine Theory, 23(3), pp. 191-20.
- [5] Waldron, K. and Kinzel, G. (2003), "Kinematics and Dynamics of Machinery." Ed.: John Wiley.
- [6] Moon, C. H. (1961), "Cam design." A manual for engineers designer and draftsman. Commercial Cam Division, Emerson Electric Company.
- [7] Flores, P. (2010), "A computational approach for cam size optimization of disc cam follower mechanisms with translating roller followers." ASME Journal of Mechanisms and Robotics, Vol. 5(4).
- [8] Zayas Figueras, E. E. (2001), "Aportación al estudio de levas desmodrómicas." Tesis doctoral. Universidad Politécnica de Cataluña.
- [9] Serrano Muñoz, A. M. (2010), "Diseño, análisis y simulación de levas planas. Aplicación informática CAM-DAS." Proyecto fin de carrera. Universidad Carlos III de Madrid.
- [10] Jiménez Abad, A. R. (2013). "ENGRANA. Aplicación Android para el cálculo de engranajes" Proyecto fin de carrera. Universidad Carlos III de Madrid.
- [11] Núñez Ruiz, C. (2013). "Aplicación Android para el estudio de ejes a fatiga" Proyecto fin de carrera. Universidad Carlos III de Madrid.
- [12] Álvarez Sanz, E. (2013). "Aplicación informática sobre androide: Blafa. Aprendizaje y cálculo de rodamientos" Proyecto fin de carrera. Universidad Carlos III de Madrid.
- [13] Shigley, J. E. and Uicker, J. J. (1995). "Teoría de máquinas y mecanismos." Ed.: Mc Graw Hill.
- [14] Chen, F. Y. (1982) "Mechanics and Design of Cam mechanisms." Ed.: Pergamon Press Inc. U.S.A.
- [15] Cardona, S. y Clos, D. (2001) "Teoría de Máquinas" (Edición en Castellano). Ed.: UPC. Barcelona.



- [16] Llamazares Aparicio, E. (2001), “Aplicación para el Cálculo y la simulación cinemática de levas planas de rotación”. Proyecto fin de carrera. Universidad Carlos III de Madrid.

8.2. OTROS RECURSOS UTILIZADOS

- [17] Apuntes de la asignatura “Cinemática y dinámica de máquinas”. (Curso 2014-2015) Departamento de Ingeniería Mecánica. Universidad Carlos III de Madrid.
- [18] Apuntes de la asignatura “Teoría de máquinas.” (2001). Universidad Politécnica de Cataluña.

8.3. PRINCIPALES REFERENCIAS A LA WEB

- [19] Ventas Android (último acceso, 14-06-2016).
<http://andro4all.com/2016/01/smartphones-android-aplasta-en-ventas-iphone-ios-couta-mercado>
- [20] iOS vs Android (último acceso, 14-06-2016)
<http://es.gizmodo.com/ios-versus-android-asi-se-comparan-en-2015-1700516382>
- [21] Tablets Android (último acceso, 14-06-2016)
<http://www.xataka.com/tablets/gartner-en-2015-las-tablets-superaran-a-los-ordenadores-se-extiende-el-dominio-android>
- [22] Sistema Audi Valvelift (último acceso, 14-06-2016)
<http://www.audi.es/es/web/es/innovacion-audi/tecnologia/sistema-audi-valvelift.html>
- [23] MIT AppInventor (último acceso, 14-06-2016)
<http://appinventor.mit.edu/explore/>
- [24] Google Material Design (último acceso, 14-06-2016)
<https://www.google.com/design/spec/material-design/introduction.html#>



ANEXO I. Manual de la app “Cálculo de levass”.

En el presente anexo se trata de dar una visión completa de todas las pantallas que podemos encontrar en la aplicación. Para ello se plantean dos pequeños ejercicios de diseño de mecanismo leva-seguidor de diferentes tramos. Se mostrará los pasos a seguir para configurar una leva con seguidor puntual y otra con seguidor de cara plana.

Al iniciar la aplicación nos aparecerá la pantalla inicial, donde tras unos segundos dará paso al menú “Home”.



Figura 115. Pantalla inicial “Cálculo de levass”.

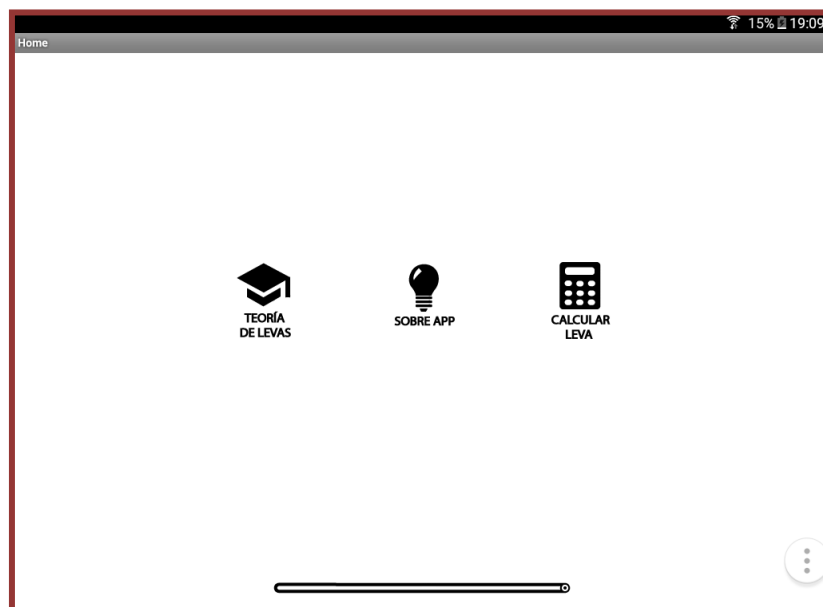


Figura 116. Pantalla “Home”.

Desde esta pantalla principal podemos acceder a los tres grandes módulos de la aplicación. A continuación mostramos las pantallas “Teoría de levas” y “Sobre App”.

Por un lado, en pulsando sobre “Teoría de levas” encontraremos unas pinceladas de la teoría básica de los mecanismos leva-seguidor que nos ayudará a comprender como funcionan estos sistemas, sirviendo de ayuda para completar un ejercicio de forma satisfactoria.

Teoría de levas
15% 19:11

DEFINICIONES

Leva

Elemento mecánico que sirve para impulsar por contacto a otro elemento mecánico, llamado palpador o seguidor, para que desarrolle un movimiento especificado.

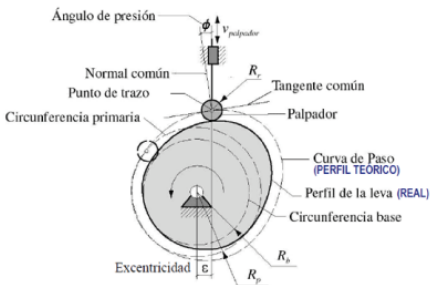
Mecanismo de leva

Sistema mecánico, con dos movimientos móviles en contacto (leva y seguidor), unidos a un bastidor, que transforma el movimiento según una ley de dependencia, denominada ley de la leva..

Ley de la leva

Es la función que relaciona el desplazamiento de la leva, lineal o angular, y el desplazamiento del palpador, lineal o angular. Viene dada por el perfil de la leva y se puede representar a través del diagrama de desplazamientos, a partir del cual se determina el perfil de la leva.

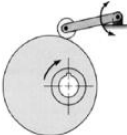
NOMENCLATURA



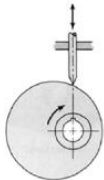
CLASIFICACIÓN MECANISMOS LEVA-SEGUIDOR

Tipo de movimiento del seguidor

· Seguidor rotacional

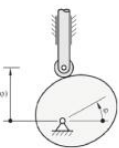


· Seguidor traslacional



Tipo de leva

· Leva radial



· Leva axial

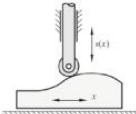
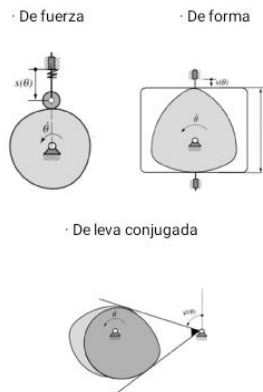


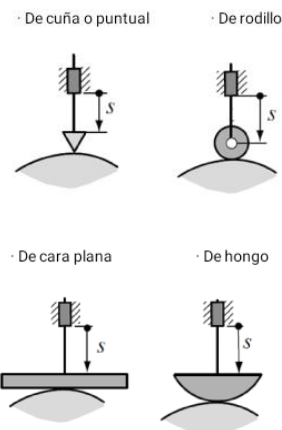
Figura 117. Pantalla “Teoría de levas”.1ª Parte.



Tipo de cierre o junta



Tipo de seguidor

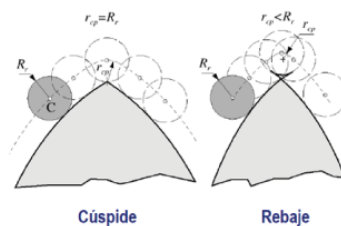


CRITERIOS DE SELECCIÓN DE LEVA

Por un lado, la ley o función de la leva debe cumplir la ley fundamental de la continuidad, es decir, las funciones de desplazamiento, velocidad y aceleración deben ser continuas.

Se admitirá una función de sobreaceleración discontinua, pero esto ocasiona vibraciones que serán más graves cuanto más elevada sea la velocidad de la leva. Además, los valores pico de velocidad y aceleración deben ser los más bajos posibles, para minimizar el consumo energético y los esfuerzos dinámicos.

Por otro lado, se ha de tener presente el efecto del radio de curvatura. Éste indica la presencia de rebajes y cúspides, fenómenos a evitar.



Además, el valor del ángulo de presión debe ser pequeño para evitar el efecto cuña que se produce entre el palpador y su guía. En la práctica, con palpadores de traslación, no se deben superar los 30° , y con los oscilantes son aceptables ángulos de presión de hasta 35° .

Nota:

Para corregir un ángulo de presión elevado, se recomienda aumentar el radio base de la leva o el radio del palpador circular, así como modificar la excentricidad.

Figura 118. Pantalla "Teoría de levas".2ª Parte.



Figura 119. Pantalla "Teoría de levas". 3ª Parte.

Por otro lado, accediendo a "Sobre App" nos encontraremos con un pequeño manual de uso de la aplicación, donde el usuario podrá ver cómo configurar los tramos de movimiento del seguidor y leer sobre las funciones más atractivas a la aplicación, como las opciones del ajustado de las gráficas, el botón "Guardar" o el botón "Exportar".

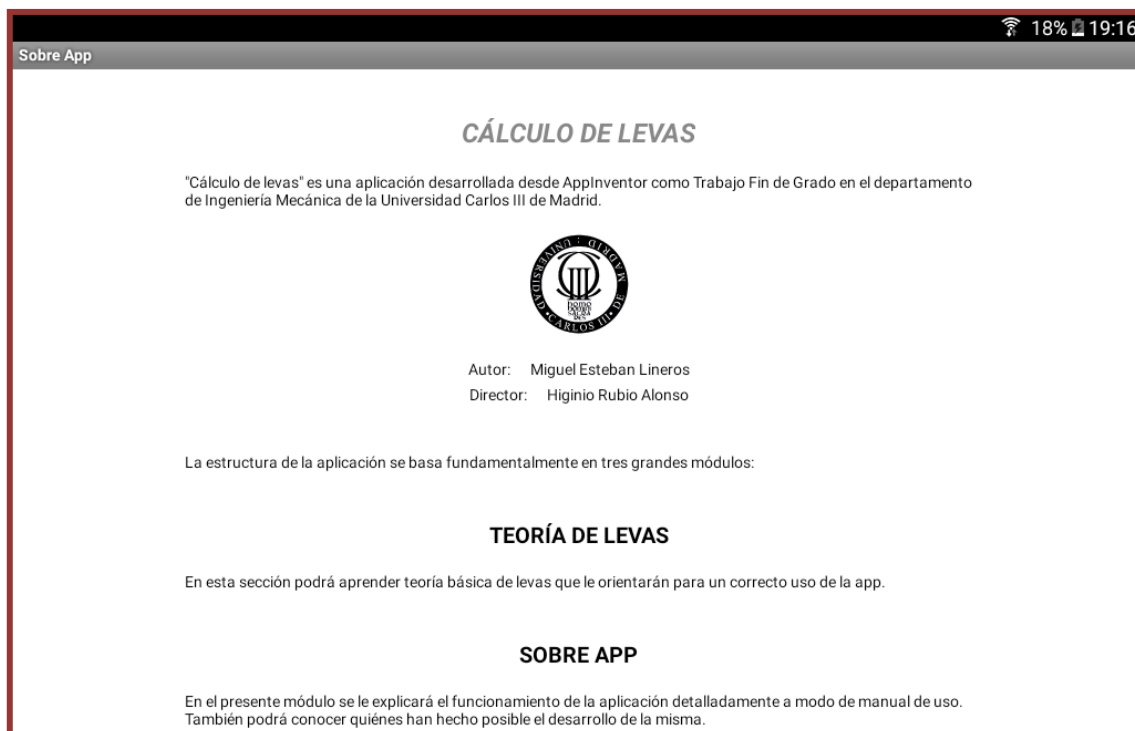


Figura 120. Pantalla "Sobre App". 1ª Parte.



CALCULAR LEVA

Este módulo principal de la aplicación se centra en el cálculo de levas planas con palpador translacional.

Usted podrá configurar desde el primer momento los parámetros característicos de la leva y del seguidor. Una vez hecho esto, se podrán visualizar los datos, analizarlos y tratar los resultados finales.

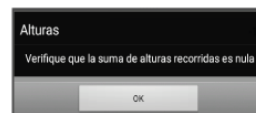
En primer lugar, seleccione el número de tramos de movimiento que tendrá su leva (limitada de 2 a 4); escoja el tipo de seguidor y en función de éste ajuste los campos de magnitudes geométricas y velocidad angular.

Seguidamente, en la configuración de tramos, se completarán todos los campos de altura recorrida, ángulo rotado y tipo de curva que desarrollará el perfil.

Ha de tener en cuenta que se debe cumplir:

- Suma de alturas igual a cero, para cumplir con la ley de la continuidad
- Ángulo total de 360°
- Tramos de detenimiento con altura nula

En caso no cumplir con algún requisito, el programa le avisará.



Una vez superada esta pantalla, accederemos a la pantalla de gráficas.

En esta pantalla podremos manipular las gráficas cambiando los factores de escala para una visualización personalizada. En caso de querer ajustar las gráficas al espacio determinado, utilizaremos la opción de "Autoajustar".

Además de visualizar las gráficas de desplazamiento, velocidad, aceleración y sobreaceleración, la aplicación nos proporciona las gráficas de ángulo de presión, para el caso de seguidor circular, y radio de curvatura, cuando se trate de seguidor circular o plano, junto a un análisis de resultados para comprobar si la leva es aceptable.

Cabe destacar a esta altura de la aplicación las opciones de "Guardar", donde se guardarán todas las gráficas en formato .png y la opción de "Exportar", donde se exportarán dos archivos .csv, uno con las coordenadas del contorno de la leva, y otro con los resultados de las gráficas S-V-A-J, pudiéndose compartir a través de aplicaciones externas.

Una aplicación atractiva de esto último puede ser el tratamiento de los datos para imprimir la leva en 3D.



Pulsando sobre el botón "Dibujar Leva" accedemos a la última parte de este módulo. Aquí podremos ver una representación de la leva diseñada y de sus curvas más características.

En el caso de elegir previamente palpador plano, el perfil de la leva será el generado por la envolvente que forma la cara del palpador para cada posición.

Figura 121. Pantalla "Sobre App". 2ª Parte.

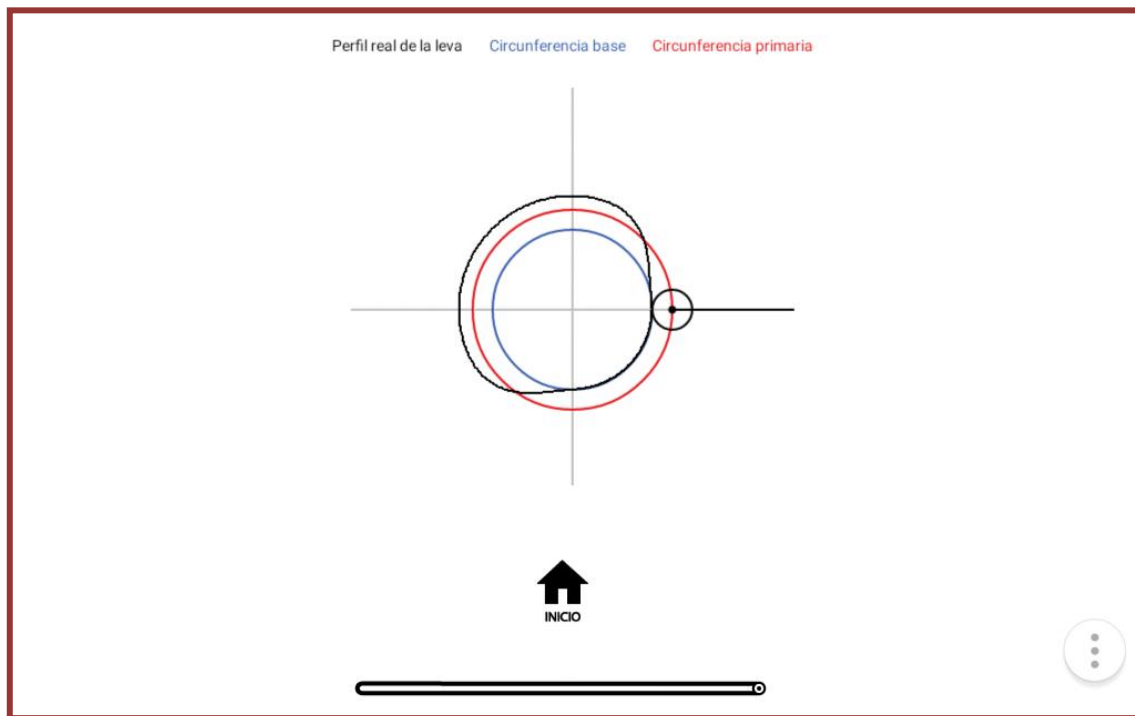
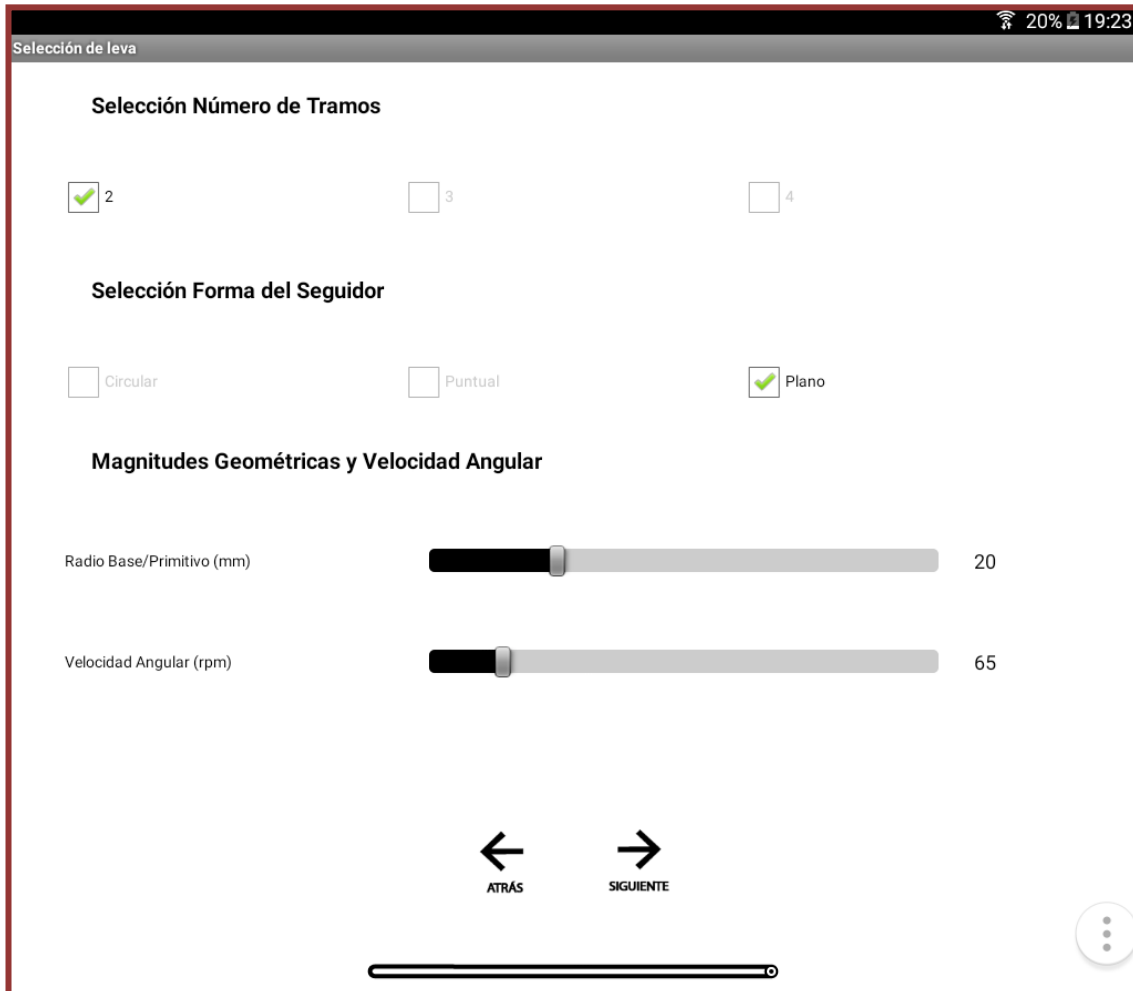


Figura 122. Pantalla “Sobre App”. 3ª Parte.



Para ver las diferentes opciones del tercer menú, “Calcular leva”, se realizan a continuación dos ejercicios donde se variará el número de tramos de la leva y el tipo del seguidor del mecanismo.

Dos tramos, seguidor de cara plana



Selección de leva

Selección Número de Tramos

☒ 2 ☐ 3 ☐ 4

Selección Forma del Seguidor

☐ Circular ☐ Puntual ☒ Plano

Magnitudes Geométricas y Velocidad Angular

Radio Base/Primitivo (mm) 20

Velocidad Angular (rpm) 65

← ATRÁS SIGUIENTE →

Figura 123. Selección de leva-seguidor plano.



Configuración de tramos

TRAMO 1

Altura recorrida (mm)

Ángulo recorrido (°)



Tipo de curva Armónica

TRAMO 2

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva Armónica

ATRÁS GRÁFICAS




Figura 124. Configuración de leva de 2 tramos.

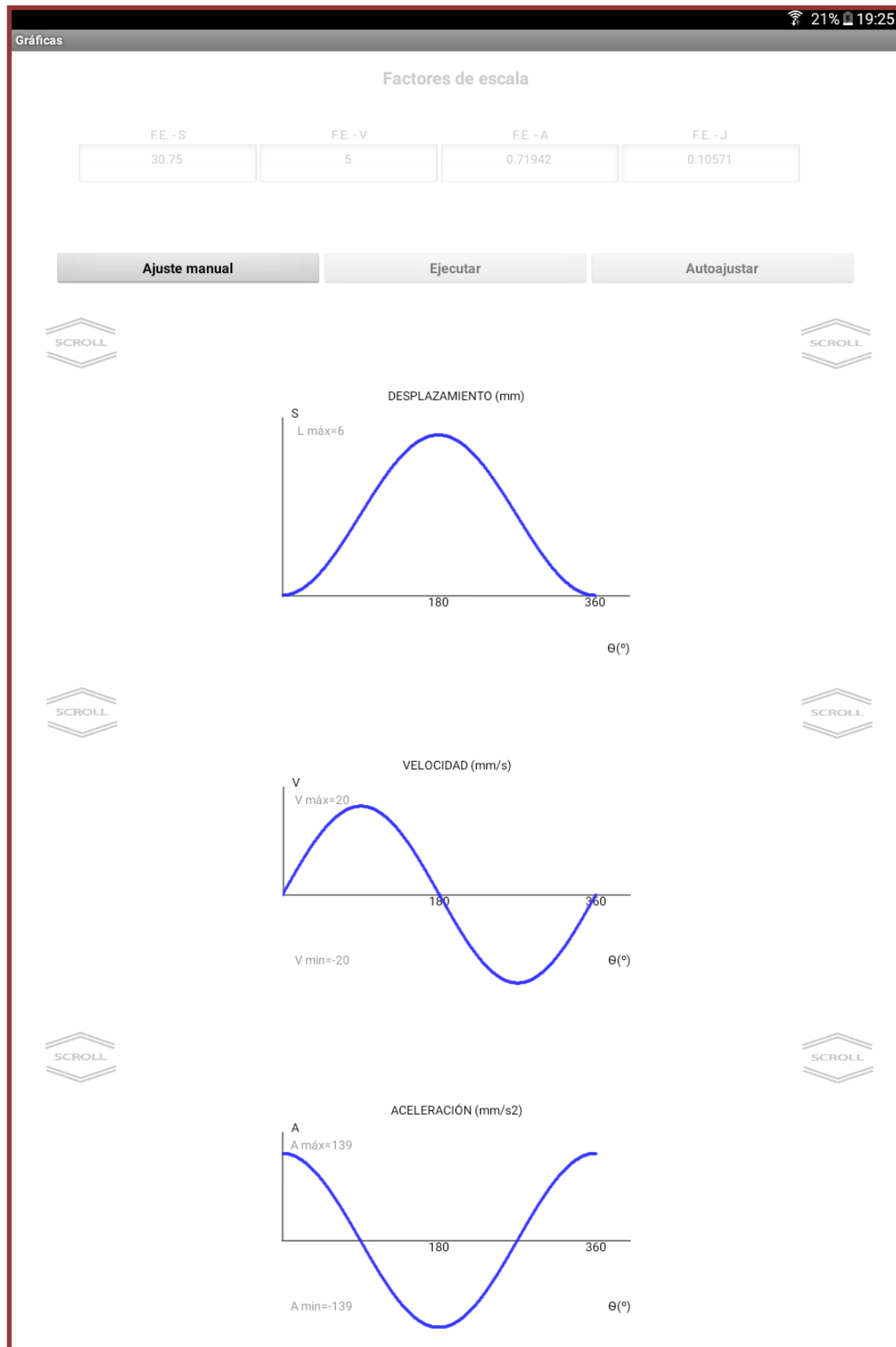


Figura 125. Gráficas seguidor de cara plana. 1ª Parte.

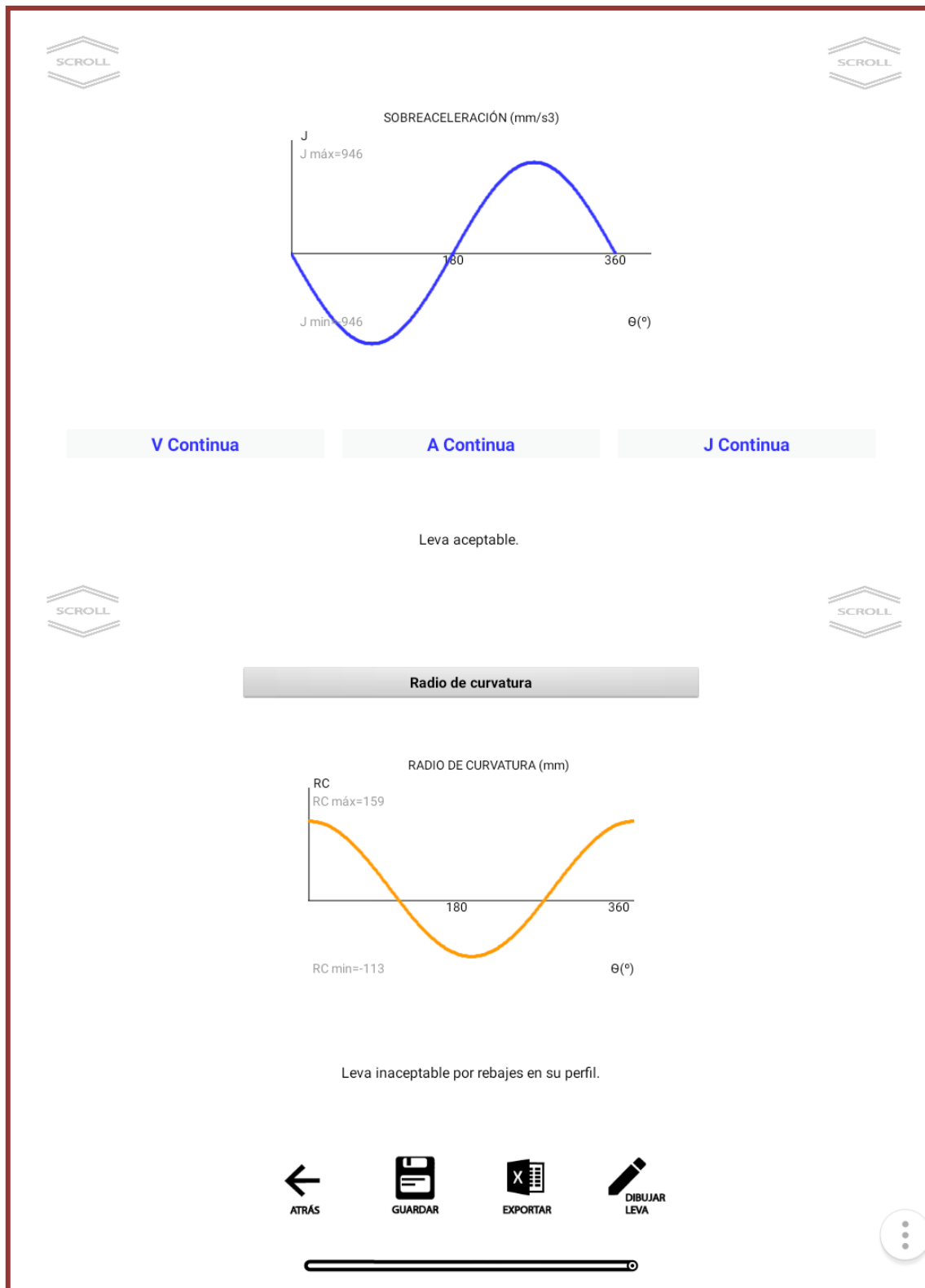


Figura 126. Gráficas seguidor de cara plana. 2ª Parte.

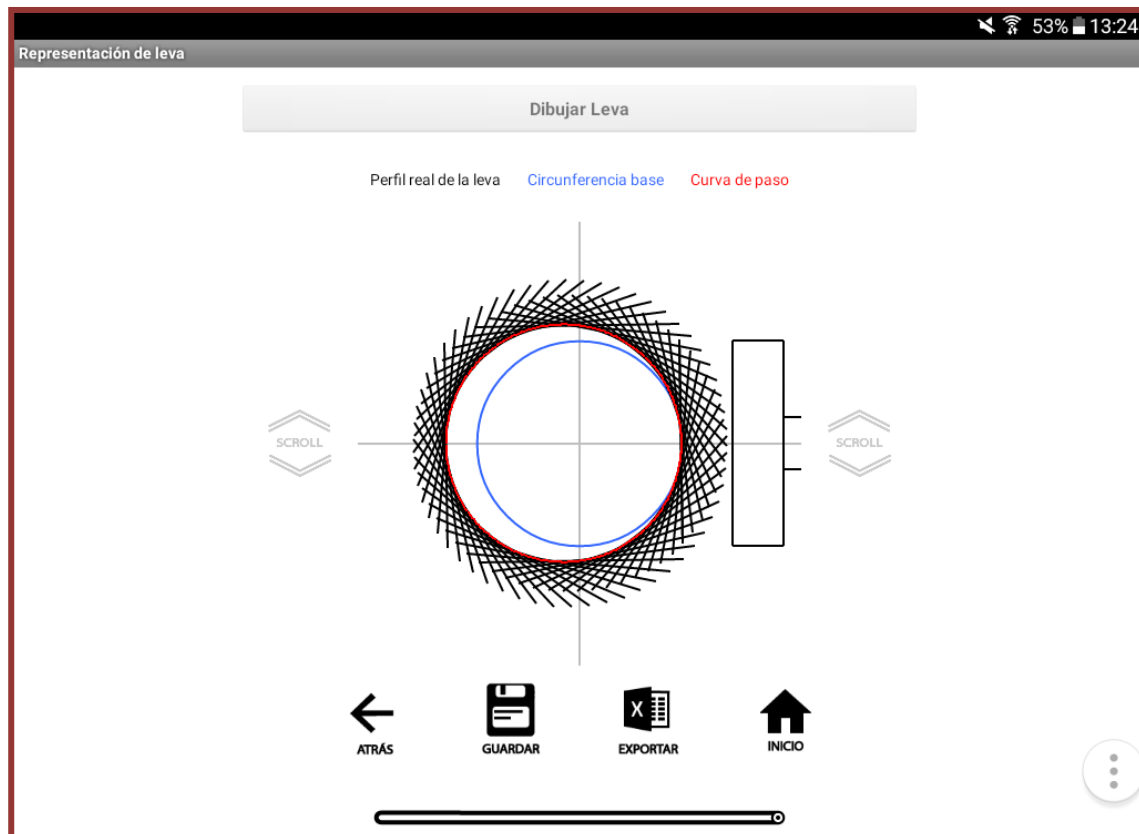
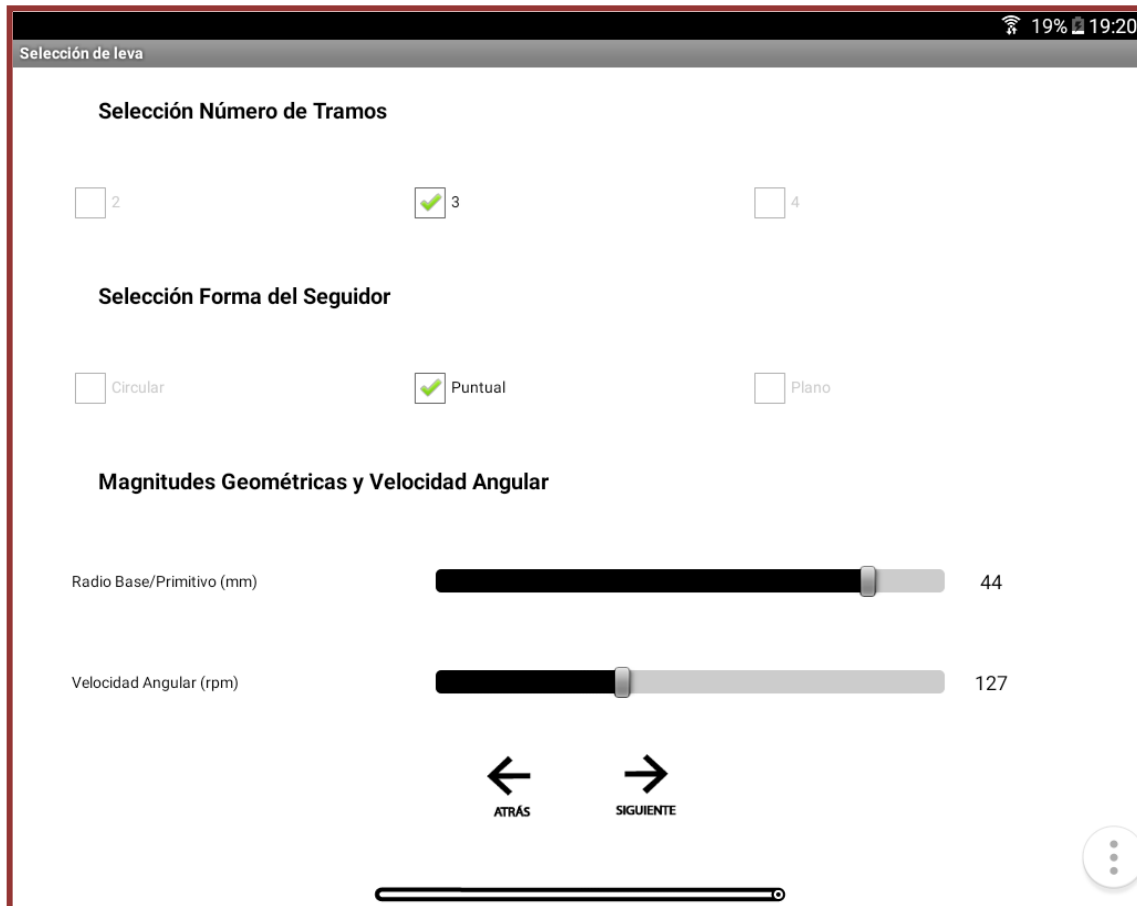


Figura 127. Representación mecanismo leva-seguidor de cara plana.



Tres tramos, seguidor puntual



Selección de leva

Selección Número de Tramos

☐ 2 ☒ 3 ☐ 4

Selección Forma del Seguidor

☐ Circular ☒ Puntual ☐ Plano

Magnitudes Geométricas y Velocidad Angular

Radio Base/Primitivo (mm) 44

Velocidad Angular (rpm) 127

← ATRÁS → SIGUIENTE

Figura 128. Selección de leva-seguidor puntual.



Configuración de tramos

TRAMO 1

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva

TRAMO 2

Altura recorrida (mm)

Ángulo recorrido (°)



Tipo de curva

TRAMO 3

Altura recorrida (mm)

Ángulo recorrido (°)

Tipo de curva

 ATRÁS  GRÁFICAS




Figura 129. Configuración de leva de 3 tramos.

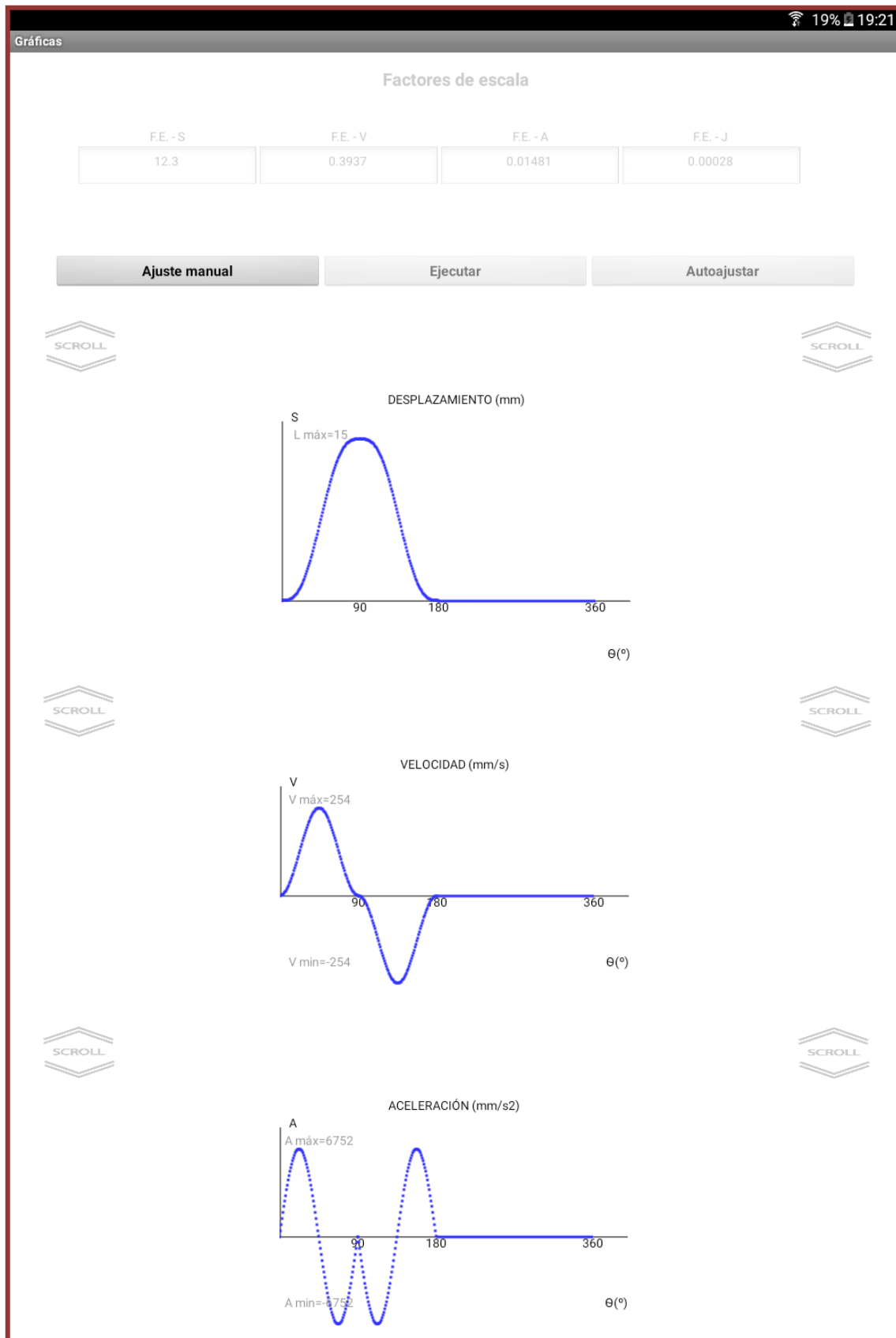


Figura 130. Gráficas seguidor puntual. 1ª Parte.

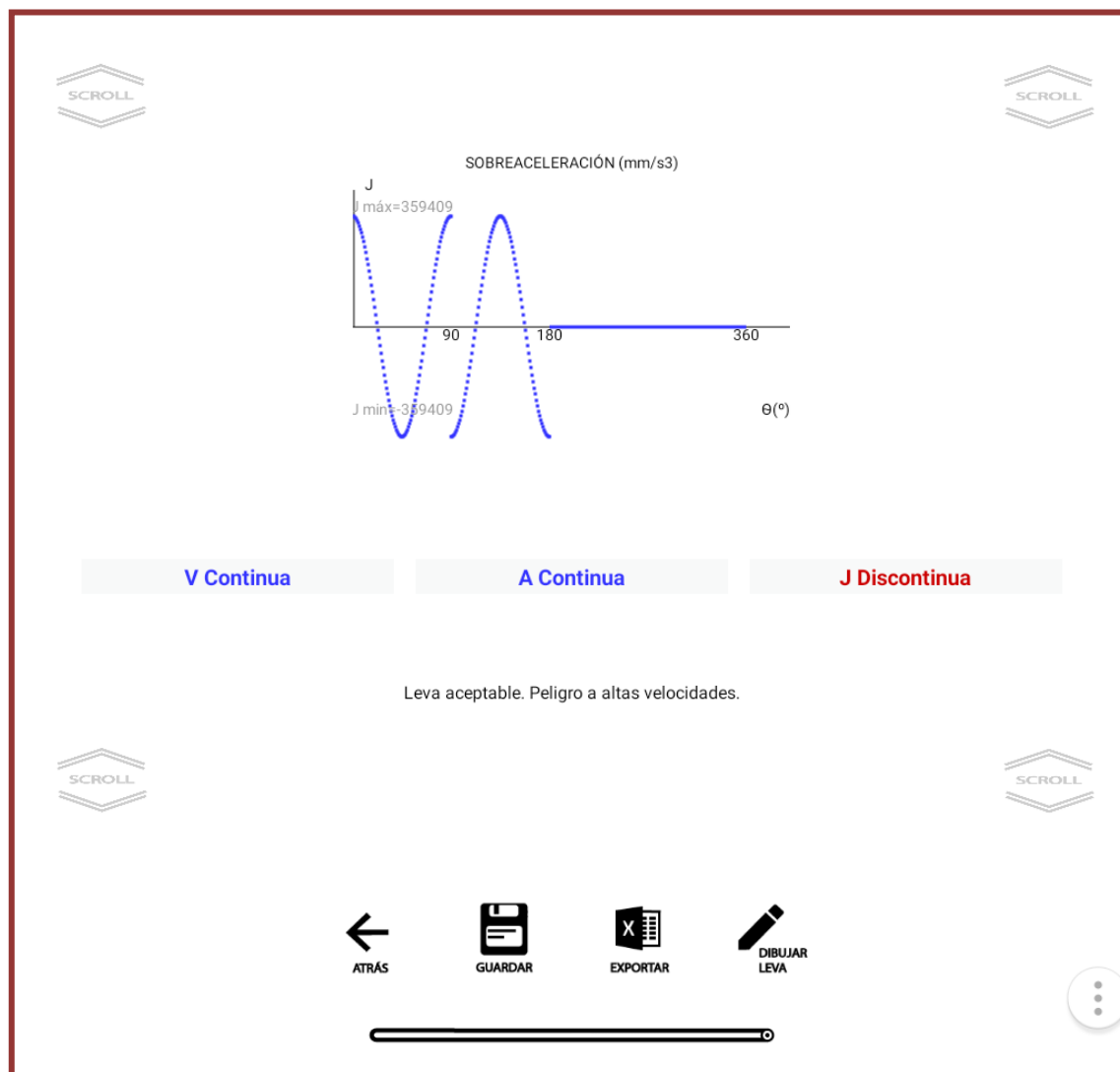


Figura 131. Gráficas seguidor puntual. 2ª Parte.

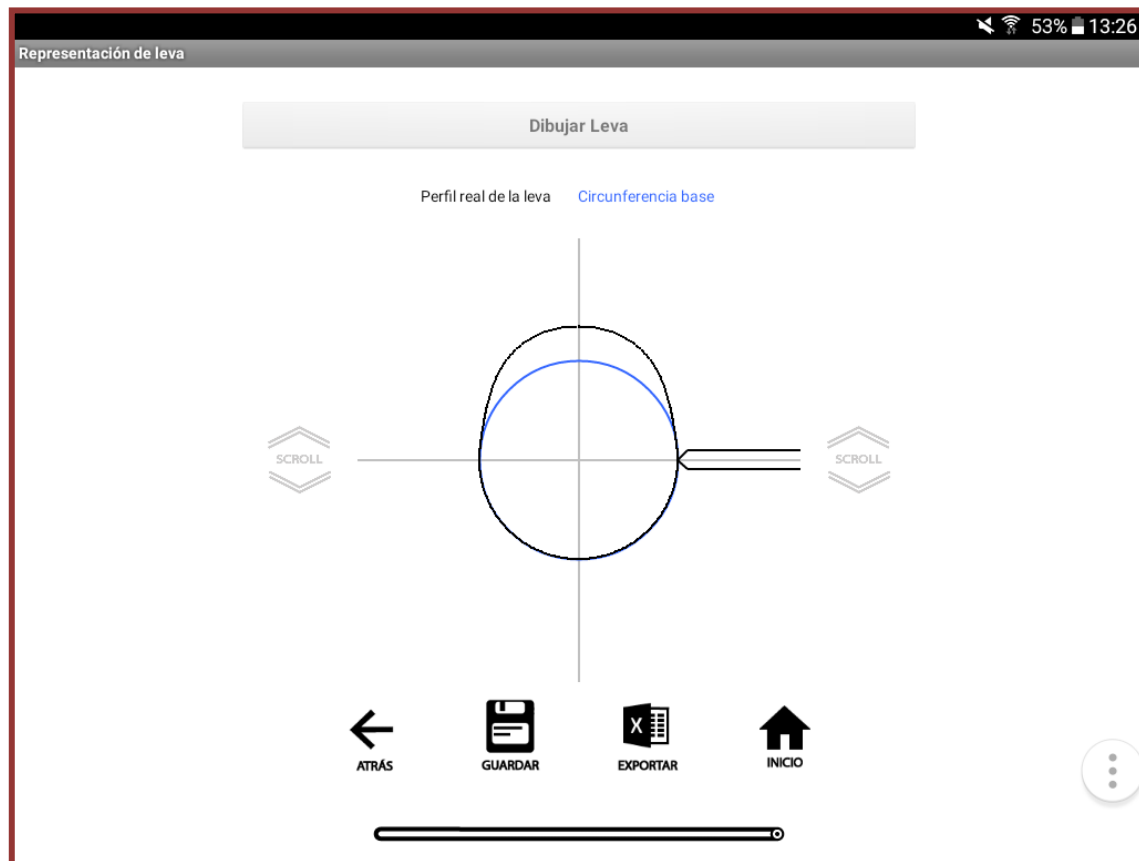


Figura 132. Representación mecanismo leva-seguidor puntual.



ANEXO II. Aplicación de prueba: Calculadora

Como primera toma de contacto con AppInventor, se decidió realizar una pequeña pero completa aplicación, una calculadora básica.

Esta aplicación es realmente importante en el proceso de estudios previos del presente trabajo porque con ella realmente entenderemos cómo funciona la plataforma elegida AppInventor. Iremos descubriendo las ventajas que ésta nos aporta y a su vez las limitaciones que nos pueden frenar.

En primer lugar se diseñó una calculadora básica en el módulo de “Designer”, donde se aprendió a cómo movernos por el entorno y cómo organizar y distribuir los componentes de nuestra app en la pantalla mediante Layouts verticales y horizontales. También se fue descubriendo poco a poco que propiedades podíamos modificar de los diferentes componentes que íbamos utilizando.

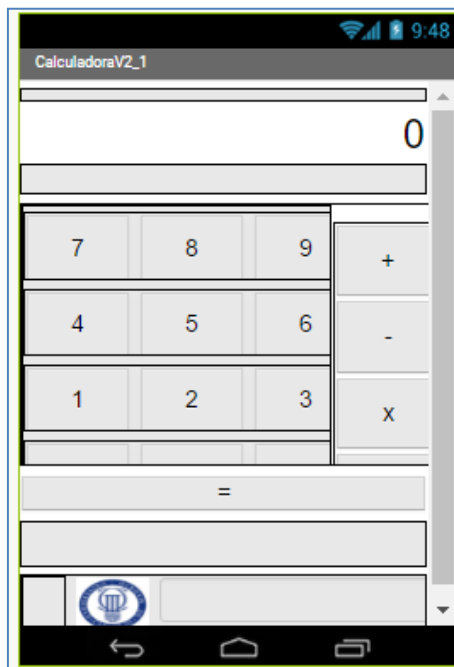


Figura 133. Visualizador de mesa de trabajo en “Designer”.

Para esta aplicación de prueba, como cabe esperar, el componente principal es el Button, un botón que al pulsarlo activa un evento y que sería utilizado como las teclas de la calculadora. Este componente sería además uno de los más importantes dentro de la app principal “Cálculo de levass”. AppInventor nos permite modificar gran cantidad de propiedades del mismo, como su tamaño, color o si va a contener una imagen o texto.

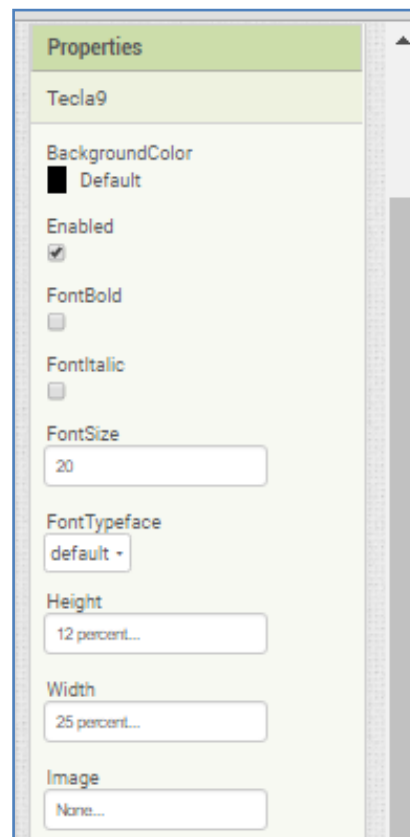


Figura 134. Detalle de propiedades de componente Button.

Otro componente que fue necesario aprender a programar para esta aplicación es el componente Label o etiqueta. Utilizaremos este componente para la muestra de datos por pantalla cada vez que pulsamos un botón, ya sea de número o de operación.

A su vez, se aprovechó esta aplicación para explorar con otros componentes de entrada de datos como los TextBox, que sería lo que utilizaríamos en la app “Cálculo de levass” para la selección de número de tramos y tipo de seguidor, y componentes para mostrar imágenes pensando también en la aplicación principal del trabajo.

Una vez plasmado en el entorno el diseño pensado, se procede a la parte programativa, en este caso desde el módulo “Blocks”. Para realizar la calculadora se pensó en una serie de algoritmos que mostrasen por pantalla el número seleccionado y se fuese almacenando en una variable interna hasta que se pulsase una tecla de operación (suma, resta, multiplicación división o potencia).

En el momento de pulsar una tecla que suponga una operación matemática, el signo aparecerá por pantalla y se procederá a guardar como variable un segundo número, independientemente de los dígitos que tenga.



Cuando se pulsase sobre la tecla “igual” se mostraría el resultado de la operación correspondiente. Además se ideó la manera de realizar operaciones encadenadas guardando el resultado en la primera variable interna y permitiendo seguir haciendo cálculo a partir de él si la tecla pulsada tras obtener el resultado era el botón “+”, “-”, “x”, “/” o “^”.

Otros botones que fueron interesantes introducir y que en términos de programación son fáciles de realizar fueron la coma para trabajar con decimales y el botón “Clear” o “C” para limpiar las variables internas y poner la pantalla en un estado inicial.

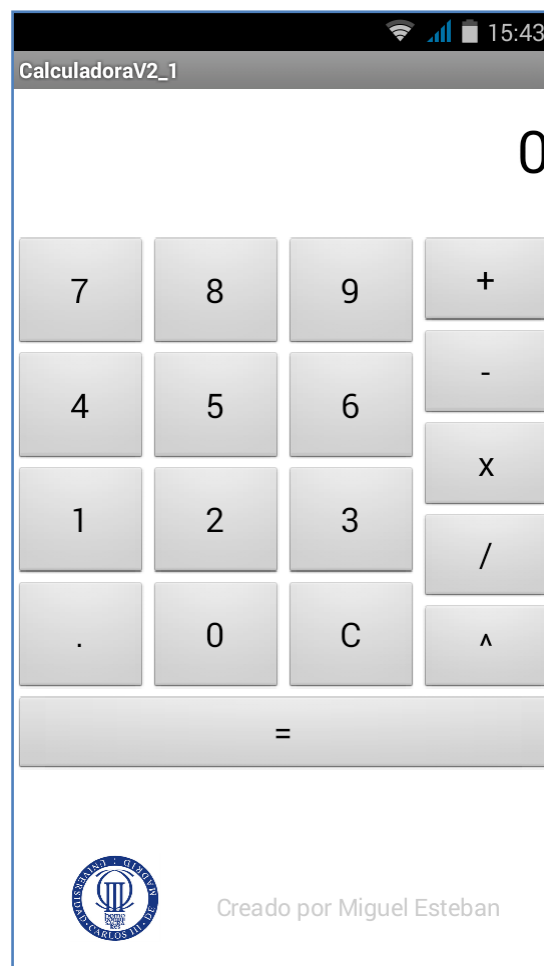


Figura 135. App “Calculadora”.

Resultó muy útil realizar esta aplicación para ver el potencial de AppInventor y lo fácil que puede resultar la programación de aplicaciones sencillas. Se pudo comprender el lenguaje de colores que usa el entorno para los diferentes bloques que se usan a modo de puzle en el módulo “Blocks”, que nos ayudarían a encarar de forma segura y efectiva el desarrollo de la app “Cálculo de Levas”.

